

Kai Priestersbach

**Web Hosting Performance Measurement.
Approach of a Web Hosting Benchmark in
Times of Virtualization**

Master's Thesis

YOUR KNOWLEDGE HAS VALUE



- We will publish your bachelor's and master's thesis, essays and papers
- Your own eBook and book - sold worldwide in all relevant shops
- Earn money with each sale

Upload your text at www.GRIN.com
and publish for free



Bibliographic information published by the German National Library:

The German National Library lists this publication in the National Bibliography; detailed bibliographic data are available on the Internet at <http://dnb.dnb.de> .

This book is copyright material and must not be copied, reproduced, transferred, distributed, leased, licensed or publicly performed or used in any way except as specifically permitted in writing by the publishers, as allowed under the terms and conditions under which it was purchased or as strictly permitted by applicable copyright law. Any unauthorized distribution or use of this text may be a direct infringement of the author's and publisher's rights and those responsible may be liable in law accordingly.

Imprint:

Copyright © GRIN Verlag
ISBN: 9783346824899

This book at GRIN:

<https://www.grin.com/document/1332069>

Kai Spriestersbach

**Web Hosting Performance Measurement. Approach of
a Web Hosting Benchmark in Times of Virtualization**

GRIN - Your knowledge has value

Since its foundation in 1998, GRIN has specialized in publishing academic texts by students, college teachers and other academics as e-book and printed book. The website www.grin.com is an ideal platform for presenting term papers, final papers, scientific essays, dissertations and specialist books.

Visit us on the internet:

<http://www.grin.com/>

<http://www.facebook.com/grincom>

http://www.twitter.com/grin_com

Web Hosting Performance Measurement

Approach of a Web Hosting Benchmark
in Times of Virtualization

MASTER THESIS

by

Kai SPRIESTERSBACH

submitted to obtain the degree of

MASTER OF SCIENCE (M.Sc.)

at

TH KÖLN UNIVERSITY OF APPLIED SCIENCES

Course of Studies

WEB SCIENCE

Collenberg, February 2023

Abstract

In this thesis, an approach for evaluating the performance of web hosting solutions is developed to facilitate the comparison of offerings from different providers and aid companies in choosing a web hosting service. The focus of the research is on identifying and investigating methodologies for collecting reproducible performance metrics within the constraints of common virtualized web hosting environments, using only PHP and MySQL functions. A comprehensive review of existing literature and research in web development and web hosting provides the theoretical basis for benchmarking web servers and measuring web performance. Industry expert interviews supplement this approach with practical experience and best practices.

The analysis of commonly used benchmarks identified in the interviews showed that well-established benchmarks and webserver load testing tools were infeasible due to the lack of command line interface access in shared and managed hosting environments. Benchmarks relying solely on PHP and MySQL functions appear promising, but they do not account for all relevant aspects and cannot incorporate external factors. External load testing tools and page speed tools were found to cover aspects not considered by server-side benchmarks, such as caching mechanisms and networking. The research also highlights the need to evaluate website loading speed and back-end administrative interface performance separately, as they are both relevant to customers but are affected differently by caching mechanisms.

The integration of these data sources is recommended for a comprehensive and meaningful analysis. However, different user groups may attach different importance to certain aspects of hosting, depending on their intended use case. Therefore, a weighted evaluation of different benchmark data for different use cases is essential, leading to the development of a multi-layered approach to benchmarking WordPress-specific hosting environments. This thesis proposes to evaluate the performance of web hosting solutions by utilizing synthetic workloads across multiple metrics to simulate real-world scenarios for improved repeatability and consistency in performance measurements. The proposed approach leads to a thorough understanding of the performance characteristics of web hosting solutions, which can be applied, evaluated, and further developed in practical settings.

Preface

After ten years, my life as a student is coming to an end. It has been a time of adventure and personal growth. It started at the age of 30, after working in digital media for more than 12 years, with a Bachelor in E-Commerce at the University of Applied Sciences Würzburg-Schweinfurt. I discovered my passion for science, which led me to the Master's programme in Web Science at the TH Köln - University of Applied Sciences. The decision to attend a master class during the COVID-19 pandemic turned out to be an invaluable experience that helped me overcome the challenges of working from home as a self-employed individual. The programme not only provided me with new insights and opportunities for my future career, but also reignited my passion for working in interdisciplinary teams - a factor that will certainly shape the way I approach my professional life in the years to come.

Acknowledgements

First and foremost, I would like to thank my wife for always being there for me. Her unwavering support and encouragement throughout my studies, despite being diagnosed with chronic lymphocytic leukaemia, has meant the world to me and has been a constant source of strength throughout this journey. I am deeply grateful for all that you have done for me and know that I would not have been able to achieve this without your help. Thank you for being my rock.

I would like to express my sincere gratitude to my primary supervisor, Irma Lindt, for her invaluable guidance and support throughout this thesis. Her expertise and mentorship were instrumental in helping me navigate the research process and successfully complete this project. I am deeply grateful for her patience, encouragement and support, which were crucial in helping me to overcome the challenges that arose along the way. I would also like to thank Irma for her timely and constructive feedback, which helped me to refine my ideas and improve the quality of my work.

I would also like to thank DeepL SE and their excellent translation tool, as well as their writing companion, both of whom were a great help in completing this thesis by facilitating efficient translations during the research and supporting the writing process. Finally, I would like to thank The Library Genesis Project and Sci-Hub, without which I would not have been able to access all the sources I needed for my research in the time and resources available. My sincere appreciation goes to the creators, such as Alexandra Elbakyan, and the operators of these platforms for their efforts to make knowledge accessible to all.

Contents

Abstract	i
Preface	ii
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Motivation for this Thesis	1
1.1.1 A Challenge for Companies	2
1.1.2 Evolution of key metrics in web hosting	3
1.1.3 Virtualization as a Game Changer	4
1.1.4 Absence of Meaningful Specifications	4
1.1.5 Cloudisation as an Amplifier	5
1.1.6 Why Traditional Benchmarking Fails	5
1.1.7 Later Commercial Use	6
1.2 Thesis Goal and Scope	6
1.3 Limitations	7
1.3.1 Dynamic Rich Media Web Pages	7
1.3.2 Specialisation in WordPress Benchmarking	8
1.3.3 Executable Benchmarks in Limited Environments	8
1.3.4 Excluded non-technical Factors	9
1.3.5 Edge Computing	9
1.4 Thesis Structure	10
1.5 Research objective and questions	11
2 Foundational Concepts	13
2.1 Definition of Web Hosting	13
2.2 Common Concepts in Web Hosting	14
2.2.1 The Web Hosting Trinity	15
Shared Hosting	15
Dedicated Server	16
Colocation	17
2.2.2 Different Hosting Architectures	17

	N-tier Architecture	17
	Load balancing	19
	Fault domains	20
2.2.3	Virtualisation	20
	Definition of Virtualisation	21
	The History of Virtualisation	21
	Virtualisation Software	22
	Virtualised Hosting	22
	Virtual Private Server	22
	Benefits of Virtualisation	23
2.2.4	Cloud Computing	25
	Cloud Hosting and Cloud Services	25
	Infrastructure as a Service (IaaS)	26
2.2.5	Content Delivery Networks	27
2.2.6	Managed Services	27
2.3	Software Concepts	28
2.3.1	Web Servers	28
	Apache	29
	NGINX	29
	PHP, SAPI, CGI and FastCGI	30
	Opcache as Performance Enhancement Extension	31
	Data Base Management Systems	31
	Key-Value Cache Stores	31
2.3.2	Control Panels	32
2.3.3	Content Management Systems	32
2.4	Web Performance Metrics	33
2.5	Network Performance	34
3	Research Design	37
3.1	Research Goals	38
3.2	Research Methods	38
3.2.1	Literature review	39
	Research on Academic Publications	39
	Research on professional literature	40
3.2.2	Expert Interviews	41
3.2.3	Benchmark Analysis	42
3.3	Challenges	43
4	Literature review	45
4.1	State of the Art	45
4.2	The Hosting Industry	46
4.3	Web Server Optimisation	47
4.3.1	Server Configuration	47

4.3.2	The PHP Workers Conundrum	48
4.3.3	Performance Measurement on NGINX	51
	Monitoring Tools	51
	NGINX as Reverse Proxy	52
	Content Caching with NGINX	52
4.3.4	Bandwidth Management	53
	Rate Limiting	53
	Connection Limiting	54
4.3.5	Load Balancing	55
4.3.6	Database Caching	55
4.3.7	CPU Sharing	55
4.4	Benchmarking in Computer Science	56
4.4.1	The Importance of Benchmarks	56
4.4.2	Specialised Web Hosting Benchmarks	57
4.5	Synthetic versus Real-User Performance Measurement	58
4.5.1	Simulating Real World Use	59
4.5.2	Understanding the PHP application stack	60
4.5.3	Discoveries from Metrics-based Assessment	61
4.5.4	Understanding Behavior of Web Servers in Stress Tests	62
4.6	Importance of Trust in Benchmarking	64
4.6.1	Limitations of Utilizing External Benchmarking	64
4.6.2	CDN Usage as an Obstacle	64
4.6.3	Variability in Website Speed Measurement	65
4.6.4	Cloud Infrastructure as a Black Box	66
4.6.5	Virtualisation and Cloudisation	67
4.6.6	Benchmark Development	68
5	Expert interviews	71
5.1	Methodology	71
	Interview Structure	71
	The Interviewer	72
	Critical and Informed Perspective	73
	Expert Selection	73
	Promoting Motivation	74
	Interview Evaluation	74
5.2	Interviewees	75
5.3	Questions	76
5.4	Summarisation of the Interviews	77
5.4.1	Interview 1: Pat	77
5.4.2	Interview 2: Max	82
5.4.3	Interview 3: Andy	86
5.4.4	Interview 4: Nip & Tuck	87

5.5	Aggregated Results	90
5.5.1	Limitations, Caching and Benchmarking	91
5.5.2	Downsides of Virtualisation	92
5.5.3	Company Size and In-House Software Development	92
6	Benchmark Analysis	95
6.1	Methodology	95
6.2	Serverside Benchmarks and Utilities	96
6.2.1	HTTP Load-Testing: Apache JMeter, Locust, Gatling	97
6.2.2	Webserver Benchmarking Utilities: Apache Benchmark and Siege	97
6.2.3	Webserver Monitoring: Nagios, Zabbix and APM	98
6.3	Website Speed Measurement	98
6.3.1	Google PageSpeed Insights	99
6.3.2	GTmetrix	100
6.3.3	WebPageTest	100
6.4	PHP-based Benchmarks	101
6.4.1	Torrisi's PHP Benchmark Performance Script	102
6.4.2	TiM International's Web Hosting Performance Benchmark	102
6.5	WordPress-based Benchmarks	103
6.5.1	WordPress Hosting Benchmark tool by Anton Aleksandrov	104
6.5.2	WPerformanceTester by Kevin Ohashi	105
6.5.3	WordPress as an Environment for Synthetic Benchmarks	106
6.6	Load-Testing and Stress-Testing Services	108
6.7	Abandoned Benchmarks	109
6.8	Client-Side Benchmarks	109
6.9	Network Performance Measurement	110
7	Synthesis	111
7.1	Results	111
7.1.1	The Need for Specialised Benchmark	111
7.1.2	PHP Benchmarking	112
7.1.3	WordPress Benchmarking Plugins	112
7.1.4	External Benchmarking	113
7.2	Comprehensive Proposal for Problem Solving	114
7.2.1	A Multi-Layered Approach	114
7.2.2	Components of a Benchmarking Framework	115
7.2.3	Frontend and Backend Performance Testing	115
7.2.4	Synthetic Performance Testing	116
7.2.5	Stress Testing	116
7.2.6	Rolling Benchmarking	117
7.2.7	Prudent Implementation	118
8	Discussion	119

8.1	Limitations	119
8.1.1	Unclear System State	119
8.1.2	Benchmark Variation	120
8.1.3	Benchmark Manipulation	120
8.1.4	Validation and Limited Testing	121
8.2	Further Research	121
8.2.1	Edge Computing as a future technology	122
8.2.2	Adjusting Metrics for Benchmarking	122
8.2.3	Further Relevant Metrics	123
8.2.4	Further Research Questions	123
9	Conclusion	125
A	Interview Transcripts	127
A.1	Interviewee 1: Pat	127
A.2	Interviewee 2: Max	161
A.3	Interviewee 3: Andy	178
A.4	Interviewee 4: Nip & Tuc	193
B	Source Codes	201
B.1	PHP Benchmark Performance Script	201
	Bibliography	203

List of Figures

1.1	Conceptual structure of the thesis. Source: Own Figure	10
2.1	Web Hosting Concept. Source: Own figure	14
2.2	Shared Hosting Concept. Source: Own Figure based on [50]	15
2.3	N-tier architecture. Source: Own Figure based on [56]	18
2.4	Load balancing. Source: Own Figure based on [56]	19
2.5	Fault domains. Source: Own Figure based on [56]	20
2.6	Virtualised Hosting Concept. Source: Own Figure based on [51]	23
2.7	On-Premise, Colocation, Hosting, IaaS, SaaS and PaaS compared. Source: Own Figure based on [53]	27
2.8	Latency and bandwidth. Source: Own Figure based on [29, Figure 1-1]	34
3.1	The Web Science ‘butterfly’. Source: [26]	37
4.1	Requests per seconds grouped by PHP worker pool size. Source: [65] .	50
4.2	HTTP request waiting times in a 32-core environment. Source: [65] . .	51
4.3	PHP application stack. Source: Own figure based on [24, Figure 1-1] .	60
4.4	Relative JIT contribution to PHP 8 performance. Source: [82]	62
4.5	The performance under varying intensities of user requests. Source: [34, p. 469]	63
6.1	Test results from Wordpress hosting Benchmarking Tool. Source: Own Figure (Screenshot)	107
8.1	WordPress Hosting – How to run WordPress on AWS. Source: Cutout from [45]	120
9.1	Multi-layered approach to benchmarking web hosting performance. Source: Own Figure.	126

List of Tables

5.1	List of interviewees.	75
5.2	Summary of the Expert Interview with Pat	82
5.3	Summary of the Expert Interview with Max	86
5.4	Summary of the Expert Interview with Andy	88
5.5	Summary of the Expert Interview with Nip & Tuck	90
6.1	Free Online Services for Website Speed Testing, excerpt and updated URLs from "Table 2" in [37, p. 123]	98
6.2	Web Hosting Performance Metrics [105]	103
9.1	Overview of the multi-layered benchmark approach	126

List of Abbreviations

CDN	Content Delivery Network
CGI	Common Gateway Interface
CLI	Command Line Interface
CMS	Content Management System
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DNS	Domain Name System
HDD	Hard Disk Drive
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
MD	Managing Director
IaaS	Infrastructure as as Service
ISP	Internet Service Provider
IP	Internet Protocol
JS	JavaScripts
KPI	Key Performance Indicator
NFV	Network Functions Virtualization
NVMe	Non-Volatile Memory Express
OS	Operating System
PaaS	Platform as as Service
PHP	PHP: Hypertext Preprocessor
QoS	Quality of Service
RAM	Random Access Memory
RUM	real-user measurement
SaaS	Ssoftware as as Service
SLA	Service-Level Agreement
SSD	Solid State Drive
SVP	Senior Vice President
TTFB	Time To First Byte
URL	Uniform Resource Locator
VM	Virtual Machine
VPS	Virtual Private SServer
WAF	Web Application Firewall
WP	WordPress
WWW	World Wide Web

Chapter 1

Introduction

In today's digital age, the importance of web hosting cannot be overstated. The web has gained significant global importance and has potentially become the most comprehensive source of information, leading to the development of various methods for individuals to communicate, share information and interact [32, p. v].

Web hosting services serve as the foundation upon which websites are built and accessed, and are thus critical components of the online ecosystem. The rapid evolution of web technologies and the increasing reliance on websites for various applications has further increased the importance of web hosting. From small personal blogs to large-scale e-commerce platforms, websites play a central role in the communication and exchange of information, goods and services. Ensuring the reliability and stability of these websites through effective web hosting is therefore essential to the smooth functioning of the digital economy. However, the choice of web hosting provider can have a significant impact on the performance and reliability of a website, as well as its ability to scale and adapt to changing user demands.

However, identifying a web hosting provider and product that adequately meets a company's needs can be a challenging task, even for those with advanced technical skills, like the following quote highlights: "In any complex system, a large part of the performance optimization process is the untangling of the interactions between the many distinct and separate layers of the system, each with its own set of constraints and limitations." [29, p. 165]

1.1 Motivation for this Thesis

Web hosting plays an important role in ensuring the accessibility and functionality of all websites. Therefore, web hosting services must be carefully considered when developing and maintaining websites. Its importance lies in its ability to provide the necessary infrastructure and support for website operation and accessibility, and its role will only continue to grow as the digital landscape evolves.

However, choosing a web hosting company can be challenging due to the numerous options available and the bold claims made by each provider in terms of speed,

reliability and service [31, p. 27]. While it is not possible to accurately determine the total number of web hosting companies worldwide, it is likely to be in the hundreds of thousands or even millions, according to [31, p. 27]. According to earlier research, it can often be difficult to differentiate between them [31, p. 27]. With so many options available and the needs of different businesses varying, it is not uncommon for businesses to face challenges in finding a hosting solution that meets their specific requirements. With the increasing importance and cost of digital infrastructure, it is becoming increasingly important for businesses to consider the different web hosting options available on the market.

This is not a trivial task, however, as it is complicated by the following factors:

1.1.1 A Challenge for Companies

The process of finding the most suitable hosting provider and product involves evaluating a number of different factors, including the resources and features offered, the level of support provided and the overall cost of the hosting service. However, comparing hosting services from different web hosting companies is made even more complex by the different nature of each provider's offering. This is further complicated by the variety of hosting plans available, which often include a range of differently specified resources such as physical CPU cores, RAM and disk space, but these resources have a significant impact on the performance of a website, so it is important to ensure that the hosting plan chosen provides sufficient resources to meet the needs of the website. As a result, choosing a hosting service is a multifaceted process that requires careful consideration of the resources and features offered by each provider.

The web hosting market offers a wide range of options, with prices ranging from as low as 5 USD [110] to more than 2,000 USD per month [111]. But sorting through web hosting offers on the basis of price alone is not a reliable solution because, especially for non-standardised products and services such as web hosting, it is crucial to consider factors other than price. The most expensive option is not always the best value. As Peter Pollock wrote in the book *Web Hosting For Dummies* "Expensive doesn't mean better" [30]. Also, size is not necessarily an indicator of quality. While it may be tempting to choose the biggest, most well-known company, this may not always be the best decision. Pollock found that smaller web hosts often offer better service, as the larger ones can become cumbersome and impersonal, while smaller hosts tend to be more attentive and personal [30]. A proper selection is especially valuable given the high switching costs in the hosting business [21, p. 1]. This means, that without a comparability of offered web hosting plans, suppliers cannot be properly selected and evaluated, which is "one of the most critical functions for the success of an organization." [3] and this process is generally complex with suppliers being evaluated on "several criteria such as pricing structure, delivery (timeliness and costs), product quality, and service (i.e. personnel, facilities, research

and development, capability, etc.).” [3] This highlights the significance of making an informed decision when selecting a web hosting service.

1.1.2 Evolution of key metrics in web hosting

Twenty years ago, evaluating web hosting services was a relatively straightforward process that involved comparing offers based on a few key specifications such as disk space, bandwidth and monthly traffic. In the past, when hosting static web pages, the amount of disk space available, the bandwidth of the connection and the monthly traffic included were the only considerations for customers. For example, in 2008 the team at Pingdom, a server monitoring provider, used only hosting disk space and included traffic to compare the offerings of well-known companies such as Dreamhost, Liquidweb and Hostway with their respective offerings 10 years earlier (1998) [11]. In the same year, hosting companies such as Yahoo, GoDaddy and 1&1 began removing data transfer quotas from their shared hosting plans, “retiring one of the oldest metrics in the hosting industry”. [19]

It is important to note that the concept of “unlimited” traffic offered by various web hosting providers is not entirely unrestricted. Many providers offer plans that advertise “unlimited” web space and bandwidth, but operate on an “average use” system or a “fair use policy”. This means that while a small percentage of sites may consume a disproportionate amount of resources, because the majority of sites utilize minimal or no resources [31, p. 19]. However, it is important to acknowledge that there are limitations to these so-called “unlimited” plans, as most providers include clauses in their terms and conditions that restrict excessive usage [31, p. 19]. Additionally, it is crucial to understand that web hosting encompasses more than simply storage space and bandwidth. The hosting server, a computer, must also allocate processing time and memory space to deliver web pages to visitors [31, p. 20]. Therefore, “unlimited” plans typically provide unlimited storage and bandwidth, but have limitations on processor time and memory usage [31, p. 20].

The advent of open source content management systems in the early 2000s, including WordPress, Drupal, and Joomla [60], led to an increase in dynamic web pages (also known as ‘web 2.0’ [7]) that allowed users to interact with the content. To support these types of websites, web hosting infrastructure was typically built on the LAMP (Linux, Apache, MySQL, and PHP/Perl/Python) stack, which allowed for the use of database queries to deliver unique content to different users [60]. This increased the importance of server-side computing resources, such as the type, speed, and number of CPU’s, the amount and type of RAM, the host limit (i.e., the number of sites on one server), and the availability of specific software. In addition to the traditional criteria of available disk space, server location, connection, and bandwidth, these factors became important indicators of the performance of web hosting services [30]. These metrics are theoretically comparable, but the technologies and configurations used by different providers can also affect the performance and capacity of a website. For

example, the use of different types of CPUs, web server software and/or operating systems can lead to variations in performance. This further complicates the process of comparing hosting services, as it is necessary to consider not only the resources provided, but also the underlying technologies and configurations used by each provider.

1.1.3 Virtualization as a Game Changer

The relevant metrics for web hosting services have changed over time, but the advent of virtualisation has made it difficult to accurately compare these services side-by-side. This is because hosting companies no longer use objective and widely recognised metrics to describe the services they provide. This is largely because, with full hardware virtualisation, vendors are unable to provide concrete and reliable information about the resources provided that can be compared or even replicated. As a result, it is difficult to compare web hosting services on an apples-to-apples basis.

The effective management of resources has become an important issue in the development of Internet server systems, which is why virtualisation has become an essential aspect of modern web hosting and its significance is undeniable. But, the performance characteristics of virtual resources are not standardised or uniformly defined, making it difficult to accurately compare the offerings of different web hosting companies. In order to accurately compare offers from different web hosting companies, it is therefore necessary to have more detailed information about the type and configuration of the virtual resources being allocated.

The growing trend towards virtualisation has rendered traditional key performance metrics (KPIs) effectively useless for comparison purposes, as these resources are no longer hosted on dedicated hardware, but are instead virtual resources defined by software.

1.1.4 Absence of Meaningful Specifications

It is becoming increasingly common for companies to also provide information about the capacity of their web hosting services in terms of the number of concurrent visitors that can access the hosted website simultaneously, in some cases combined with other technical specifications such as available RAM, PHP memory limit or the number of CPU cores. Today, the number of websites that can be hosted simultaneously and artificially defined *performance levels* are key factors on the pricing pages of most web hosting companies [81], [83]–[85]. While this may appear to facilitate comparability, it should be noted that the relevance of these figures is limited to comparisons between packages from the same provider, as system configurations may vary among different web hosting companies.

Without a standardised way of measuring web server performance, it is very difficult to know which host offers the best value for a customer's needs. This lack of comparability makes it difficult to determine which service is best for specific needs. This highlights the need for alternative approaches to evaluating the performance and capacity of web hosting services in the context of virtualised resources.

1.1.5 Cloudisation as an Amplifier

The ongoing cloudisation of hosting infrastructure, built on an Infrastructure-as-a-Service environment, exacerbates this problem: "Cloud service providers often define host tiers by the allocated resources, but the differences in the underlying hardware, architecture and performance tuning can result in varying capabilities even between similar configurations." [42]

It has been recognised that determining the appropriate type of hosting for a web application can be a significant challenge [48, p. 233] due to the different solutions offered. It has also been observed that the hosting landscape is undergoing significant changes, with an increasing number of individuals exploring alternative hosting options such as cloud hosting, as opposed to the more conventional methods of dedicated and shared hosting [48, p. 234]. But according to [38], enterprises that are considering migrating their IT systems to the cloud often have a limited understanding of the infrastructure that is being offered to them, as they only have a *black box view* [38]. While it is possible to find information about server pricing and specification, there is limited information available about the performance of cloud infrastructure [38].

This means that establishing objective criteria and metrics that can be practically replicated is further complicated by the highly dynamic allocation of resources based on current application needs in modern hosting infrastructures, as resources in virtualised servers are managed by autonomous controllers consisting of a monitor to measure workload and performance metrics, a predictor to forecast future workload conditions, and a resource allocator to calculate the required capacity [20]. As a result, any metric used to quantify performance is highly dependent on the current state of the resource control mechanisms.

1.1.6 Why Traditional Benchmarking Fails

Although, "Benchmarking is a widely used method in experimental computer science, especially for comparative evaluation of tools and algorithms", [55], traditional benchmarks cannot be run on shared or managed web hosting environments, as they are only allowed to run pre-installed software and modules for security and stability reasons.

In addition, it is uncertain to what extent the results of synthetic system benchmarks accurately reflect the actual performance of a web server: “Ultimately, [with benchmarking], you’re trying to get as close to real-world usage and workload so that you can make your architecture decisions in mind”, according to [57], which implies that there is a need for a specialised benchmark for evaluating web hosting environments.

Existing web hosting benchmarks are examined in more detail in *chapter 6*.

1.1.7 Later Commercial Use

The development of a new type of benchmark, based on real-world scenarios and providing greater market transparency and comparability, could form the basis of a new type of web hosting comparison service in the future. A practical solution could improve comparability and transparency in the web hosting market, enabling users to make more informed decisions based on their specific needs and objectives when selecting providers.

Developing a reproducible and objective method for measuring web server performance on shared resources in virtualised environments could be a significant competitive advantage for a web hosting comparison site.

1.2 Thesis Goal and Scope

As defined by the Cambridge Advanced Learner’s Dictionary & Thesaurus benchmarking is “the act of measuring the quality of something by comparing it with something else of an accepted standard” [88]. It has become evident that a lack of standards exists in the web hosting market, making it difficult to compare and evaluate offerings from different providers. In light of this, it is imperative to develop a methodology for evaluating the performance of web hosting solutions to aid in the decision-making process for companies seeking to choose a web hosting service.

This thesis will therefore examine the various aspects of web hosting, including the different types of web hosting services, the technologies and infrastructures involved, and the important factors to consider when benchmarking web hosting providers. In addition, existing benchmarks are examined for their applicability.

By developing a practical approach to evaluating web hosting services, including virtualised hosting, this thesis has the potential to make a significant contribution to the field of experimental computer science. The aim is to provide a theoretical technological foundation that can be applied to all web hosting services, enabling researchers to accurately and consistently measure and control resources in a reproducible manner. The ultimate goal is to facilitate the production of scientifically valid experimental results in this area.

1.3 Limitations

In academic research, it is often necessary to use specific benchmarks to accurately and effectively evaluate a particular system, process or phenomenon. This is because a specific benchmark is designed to test particular aspects or characteristics of the object of study and thus provides more focused and relevant data. In contrast, a generic or general benchmark may not adequately capture the nuances or specifics of the system or process being assessed, resulting in less accurate and potentially misleading results. Therefore, the creation of a targeted benchmark will allow for a comprehensive and scientifically sound evaluation of performance, resulting in more dependable and valuable outcomes. However, the scope of the benchmark must be specific and appropriately balanced. It should be significant enough to hold scientific importance and commercial interest, but also narrow enough to be thoroughly covered within the confines of this thesis.

Consequently, the following constraints apply:

1.3.1 Dynamic Rich Media Web Pages

The evolution of the Web has resulted in three distinct classes of experiences: Hypertext documents, rich media web pages, and interactive web applications [29, p. 166]. Hypertext documents are the foundation of the World Wide Web. Essentially plain text documents with support for hyperlinks and basic formatting [29, p. 166]. This initial definition of hypertext was later extended to support additional hypermedia resources (such as images and audio) and added other primitives for richer layouts [29, p. 166] allowing the creation of rich media web pages. Adding dynamic functionality with JavaScript, later revolutionised by dynamic HTML and AJAX, transformed web pages into interactive web applications [29, p. 166]. From a performance perspective, each of these classes requires a unique approach to communication, metrics and performance definition [29, p. 166].

This means that benchmarking the performance of static hypertext documents is different from benchmarking rich media websites and from measuring the performance of web applications. Benchmarking the performance of static hypertext documents requires different techniques and metrics than those used to measure the performance of rich media websites and web applications. When benchmarking static hypertext documents, the focus is on page load time, whereas when benchmarking rich media websites and web applications, the focus is on user experience, such as the time it takes a user to complete a task or the responsiveness of the application [29, p. 170]. This distinction must be taken into account when evaluating the performance of hosting solutions for interactive web applications.

Today, the majority of websites use CSS (Cascading Style Sheets are used by 96.8% of all websites [116]), images, JavaScript libraries (82.2% [114]) and other media along with the pure hypertext documents to have rich visual layouts with different

media types, which are in most cases generated by a content management system (>67.8% [112]). In terms of server-side programming languages, PHP is the most widely used, with an estimated 77.8% of all websites using it, according to data from Q-success [115].

Accordingly, the main focus of this thesis is to assess the performance of web hosting solutions for dynamic rich media websites.

1.3.2 Specialisation in WordPress Benchmarking

As can be seen from the introduction, it is important to have a benchmarking solution that is specifically designed to measure the performance of a particular type of website in order to accurately assess its performance and identify any potential problems. From a commercial point of view, the potentially largest market would be the most interesting. According to Q-success data, WordPress is the most popular content management system and is used by 43.1% of all websites, giving it a 63.6% share of the content management system market [113]. This high market share means that a significant proportion of websites are built using WordPress, making it an important platform to consider when developing a benchmarking solution. By developing a WordPress-only benchmark, it may be possible to meet this demand and provide those working with WordPress with a valuable tool that is more likely to be successful in terms of monetisation. In addition, WordPress is a complex and feature-rich CMS that is used to build a wide range of websites, from simple blogs to large e-commerce stores.

Consequently, the decision to focus on the development of a benchmark for WordPress only is driven by the significant market share and importance of the platform, as well as the need for a specialized solution designed to accurately measure the performance of WordPress-based websites.

1.3.3 Executable Benchmarks in Limited Environments

For future commercialisation, it is important to ensure that the benchmarking solution is compatible with as many different hosting environments as possible. To ensure maximum compatibility, it is necessary to avoid using methods that may not be supported in all hosting environments. One such method is the use of program code that is compiled to run on specific hardware architectures. Compiled code is used to create efficient and high performance applications, but execution may not be available in all hosting environments, especially those that do not offer Command Line Interface (CLI) access. Typically, web hosting environments do not provide this type of access for a number of reasons, explained in detail in *chapter 2*. By avoiding the use of binary code and other methods that may not be widely supported, the benchmarking solution can be made more compatible with a wider range of hosting environments. Additionally, Colocation and dedicated servers are not included in

the web hosting comparison as these can be compared on the basis of hardware specifications and the range of possible software configurations is too wide to be influenced by the hosting provider.

In light of this, the scope of this thesis is limited to server-side, user-level benchmarks that can be run on any hosting service capable of running the WordPress CMS (PHP and MySQL/MariaDB) without root privileges or access to the command line interface.

1.3.4 Excluded non-technical Factors

In addition to the basic service of hosting websites, web hosting companies often offer a range of additional features and support options, such as backup services and security measures, which can have a significant impact on the overall value of their hosting packages. Of course, service level agreements (SLAs), maintenance contracts, response times, support staff competence and accessibility are also important factors to consider when choosing a web hosting provider. The presence and quality of these additional services should be taken into account when evaluating the offerings of different hosting providers, as they can greatly affect the reliability and security of the hosting service, and thus the overall value of the service. These considerations should be included in a transparent web hosting comparison service in order to provide a fair and accurate evaluation of the offerings of different hosting providers. However, these factors are not considered in this paper.

As a result, the focus of the thesis is on the technical aspects of the topic, thereby excluding non-technical factors to maintain a manageable scope and ensure completion within a reasonable timeframe.

1.3.5 Edge Computing

In recent years, a novel approach to computing has emerged, known as edge computing. This approach entails the provision of limited computing power directly at the Content Delivery Network (CDN) level. This enables websites to process real-time data generated by users, resulting in an enhanced user experience. According to [92], the field of edge performance benchmarking has gained significant attention in the last five years. [92] have also noted that edge computing is seen as a key development in the future of the Internet, as it will use resources located close to users, sensors and data storage to provide more efficient services. They suggest that this will lead to the emergence of a vast, geographically dispersed and resource-rich distributed system that will play a major role in the future Internet [92].

Given the potential significance of this technology, it is worth mentioning. However, as it plays a minimal role in the majority of websites, it is not included in the scope of this thesis.

Colocation and dedicated servers are also not considered in this work, as these can be compared with standardised hardware and network specifications.

1.4 Thesis Structure

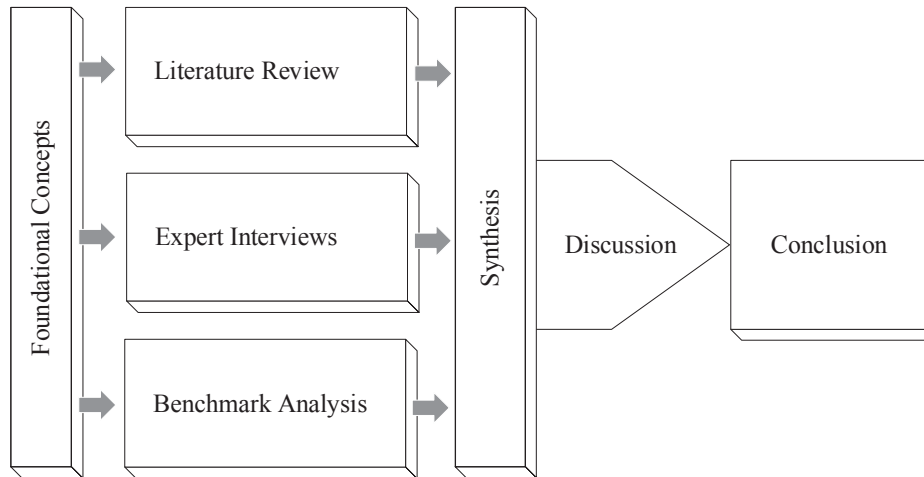


FIGURE 1.1: Conceptual structure of the thesis. Source: Own Figure

This thesis is structured as follows:

- *Chapter 2* introduces the basic concepts and terminology that are essential to understanding the research, explaining their definitions and meaning in clear and concise terms. This chapter also provides a detailed overview of the most common web hosting concepts, as well as a description of virtualisation and resource control terminology.
- *Chapter 3* provides an overview of the research design, methods and models used in this thesis. The problems addressed are also described in more detail, providing a clear rationale for the research approach chosen.
- *Chapter 4* is a review of the relevant literature and a summary of the current state of knowledge on the topic. The aim is to extract the theoretical basis of benchmarking and the conditions for evaluating web servers and web performance.
- *Chapter 5* provides a detailed explanation of the methodology used to conduct the qualitative research. This includes a detailed explanation of the procedures and techniques used to collect data through exploratory expert interviews. The results of these interviews are also comprehensively summarised in this chapter.
- *Chapter 6* provides an examination of the benchmarks identified in the research process, particularly with regard to their suitability in the context of the research questions.

- *Chapter 7* provides a synthesis of the research conducted throughout the study by combining the findings from the literature review, the qualitative research and the benchmark analyses, as shown in Figure 1.1.
- *Chapter 8* critically examines the implications of the research findings for both web scientists and web hosting professionals and provides insight into the potential applications of the findings.
- *Chapter 9* serves as a comprehensive summary of the research conducted throughout the study and provides a concise overview of the main findings, conclusions and their significance in the field of web science, as well as addressing limitations and future research directions that have been identified.

1.5 Research objective and questions

This thesis is generally concerned with a design rather than an explanatory objective. Rather than seeking a theoretical understanding and explanation of a particular phenomenon, the aim of this work is to design an artefact [54] - in this case, a novel approach to benchmarking web hosting environments. However, in order to develop a tangible solution, one must first gain a comprehensive understanding of the field, the current state of research, and existing offerings and solutions in the area of web hosting and benchmarking.

The objective of this thesis is to develop a methodology for evaluating the performance of web hosting solutions in the context of hardware virtualisation. The aim is to facilitate the comparison of offerings from different vendors and assist organisations in selecting the most appropriate web hosting service. By providing a systematic approach to benchmarking web hosting solutions, the thesis aims to improve the decision-making process for organisations seeking to select a web hosting service. The key performance indicators that have been identified as being of significant importance to customers are back-end performance and front-end load speed.

The specific scope of this thesis is to investigate and identify methodologies for the collection of reproducible and traceable performance metrics within the constraints of common web hosting environments using only PHP and MySQL functions. In summary, the aim of this thesis is to propose a methodology for evaluating web hosting solutions through benchmarking.

In order to achieve this goal, the following guiding research questions will be addressed:

- What are the key considerations when developing a benchmark?
- How are resources managed, allocated and monitored in web hosting systems?
- What performance metrics can be effectively collected in a web hosting environment?

- Which of these are traceable, repeatable and meaningful in terms of perceived website speed and backend performance?

Note: A Master's thesis is generally a significant piece of work that requires a considerable amount of time and effort to complete. In order to keep the workload manageable, it is important to focus on a specific area and to limit the scope of the research. This thesis is intended to be a starting point for more extensive research into the approach presented, which can be continued with a future research project or within a company through implementation and evaluation.

Chapter 2

Foundational Concepts

The following chapter provides an introduction to the key background information and basic concepts necessary to understand the research presented in this thesis. It serves as a starting point for the reader, introducing the context and providing the foundation upon which the subsequent chapters are built.

It is important to note that due to the lack of a standardised definition for the services offered in the web hosting industry, individual offerings may differ from the definition used in this paper. For example, providers may assemble a pool of dedicated computers but not market it as cloud hosting, even though the service is similar in nature. This highlights the need for clear and consistent terminology in the web hosting industry, which is essential for accurate and meaningful analysis and comparison of web hosting services. In addition, the web hosting industry is undergoing significant change and transformation.

2.1 Definition of Web Hosting

According to the Gartner Information Technology Glossary [98], Web Hosting is “a service in which a vendor offers the housing of business-to-business (B2B) or business-to-consumer (B2C) e-commerce websites via vendor-owned shared or dedicated servers and applications for enterprises at the provider-controlled facilities. The vendor is responsible for all day-to-day operations and maintenance of the website. The customer is responsible for the content.”

For a website to be accessible on the World Wide Web, it must be hosted on a computer that is connected to the Internet [31, p. 10], see Figure 2.1. The host computer can be any computer located anywhere in the world, it must have power, an internet connection, and a dedicated IP address [31, p. 10]. An IP address is a unique identifier assigned to each device that connects to the Internet [31, p. 10]. This means that the World Wide Web (WWW) would basically be completely empty without web hosting services.

The web hosting industry is a significant sector, with an estimated hundreds of millions of websites currently accessible on the internet [31, p. 9]. As individuals and organisations around the world establish a presence on the World Wide Web, they

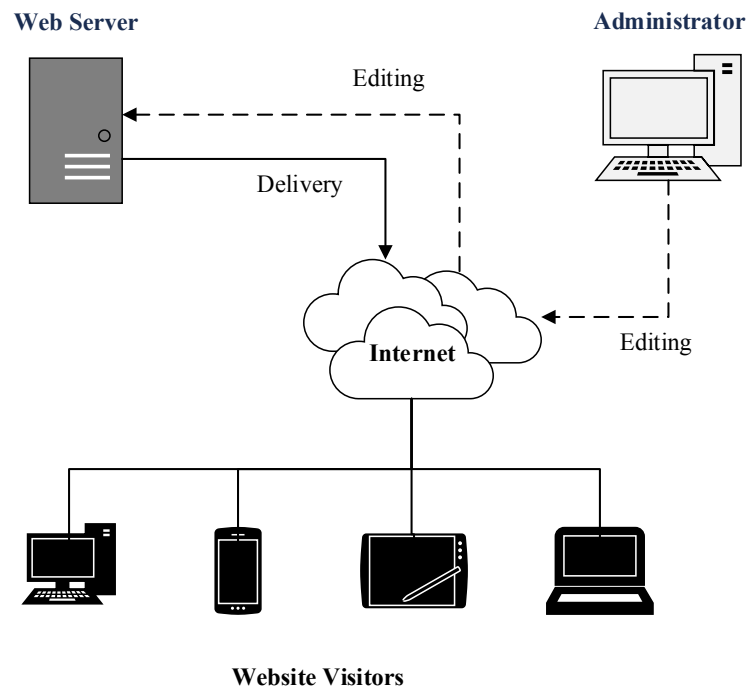


FIGURE 2.1: Web Hosting Concept. Source: Own figure

are often introduced to a new area of technology that can be initially confusing and daunting [31, p. 9]. For many individuals, understanding the concept of web hosting, its purpose and implications can prove challenging [31, p. 9].

2.2 Common Concepts in Web Hosting

Web hosting is the practice of providing storage and access to websites on the Internet. This is usually done by a third party service provider who provides the technology and infrastructure necessary to host websites and make them available to internet users. Web hosting plays a crucial role in the functioning of the Internet, as it allows individuals and organisations to publish their own websites and make them available to a global audience [118]. Web hosting services therefore provide individuals and organisations with the technology and resources needed to host a website or web-based application on the Internet.

Web hosting typically include four essential services:

1. **Server hardware and software:** A web server, operating system, and web server software are required to host a website. Some web hosting arrangements also include email hosting capabilities.
2. **Network infrastructure:** Web hosting providers also offer the network infrastructure and connectivity needed to deliver content to users.

3. **Storage and bandwidth:** Web hosting providers offer storage space for website files and data, as well as bandwidth for transmitting content to users.
4. **Technical support:** Web hosting providers often offer technical support to help customers with issues related to hosting their website or web-based application.

2.2.1 The Web Hosting Trinity

There are several different types of web hosting services available in the industry. To facilitate understanding and improve the readability of the following chapters, key concepts are introduced and briefly explained in this section.

In 2005, the hosting industry was divided into three segments based on server exclusivity: Shared Hosting, Dedicated Hosting and Colocation. [9, p. 651].

Shared Hosting

Shared hosting is a method of hosting websites in which the websites of multiple users are stored on a single server, with shared use of software and supporting hardware [9, p. 651]. Shared hosting is the most cost-effective form of hosting, as the cost of the server is shared among all customers with a shared hosting plan on that server [39, p. 19]. Therefore, shared hosting is often a suitable option for those starting out with web hosting, but the level of service can vary depending on the host, the websites hosted on the machine and the number of sites allowed by the host [31, p. 223].

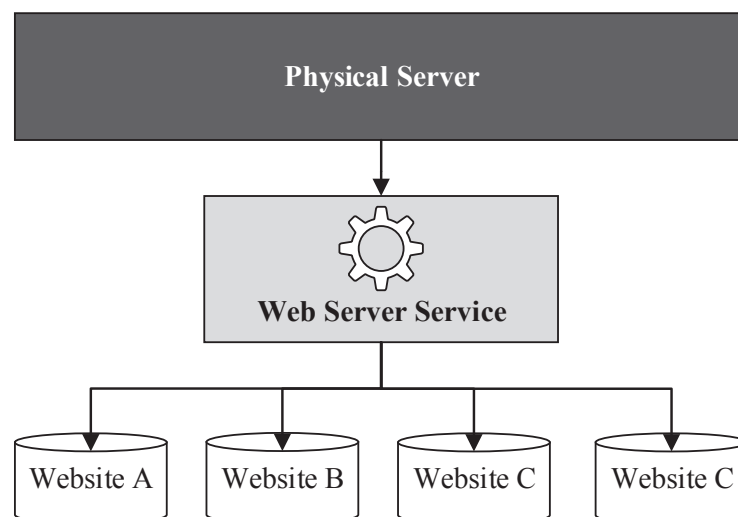


FIGURE 2.2: Shared Hosting Concept. Source: Own Figure based on [50]

All resources on the server, including memory, disk space and processor speed, are shared by all the sites on the machine [31, p. 223]. This means that if one site experiences a high level of traffic or uses a large amount of server resources, it can

negatively affect the performance of the other sites on the server [31, p. 223]. When this happens, the web hosting company may require the customer to upgrade to a more expensive server if they are consuming too many resources from the physical server, which may have a negative impact on the availability or performance of other sites hosted on the same server [39, p. 19].

Furthermore, the customer's website may be vulnerable to security issues (e.g. outdated software) caused by other websites on the same server, and the website's performance may be compromised by misbehaving sites [39, p. 19]. In most cases, there is limited opportunity to configure and tune the server, leaving the customer at the mercy of the web host's server configuration [39, p. 19].

Dedicated Server

Dedicated servers are servers that are purchased (or leased) by an individual or organisation, providing them with a guaranteed amount of resources that are not shared with other users [31, p. 224]. These servers are typically housed in a data centre and are physically exclusive to the individual or organisation [31, p. 224]. With dedicated hosting, an organisation purchases the entire resources of one or more servers from a service provider and is given full control over a fixed amount of resources, such as dedicated bandwidth, CPU, RAM and storage [48, p. 234]. The use of dedicated servers for web hosting, as shown in Figure 2.1, is the earliest method of hosting a website on the World Wide Web.

While this is the most expensive hosting option, it also offers the highest level of power and control over the hosting environment for a website [31, p. 26]: This is also one of the key benefits of using Linux for web hosting [47, p. 452]. The flexibility to customise and tweak various aspects of the kernel, the core component of the operating system, to suit specific requirements is unmatched [47, p. 452]. For example, according to [29, p. 34], optimising TCP performance has been shown to provide significant benefits for every new connection to servers, regardless of the type of application. In addition, the Linux kernel offers a large number of configurable parameters, numbering over 1,000, allowing a high degree of customisation and optimisation [47, p. 452]. These parameters can be used to fine-tune the performance, security and stability of the system and adapt it to the specific requirements of the application or service it is running [47, p. 452]. This level of flexibility and tunability is a key feature of Linux and allows it to be tailored to meet the needs of a wide range of use cases [47, p. 452]. It is also worth noting that the extra layer of virtualisation, which can have some impact on performance, is not present in this case unless the server is virtualised into separate virtual servers by the user [39, p. 21].

Colocation

The third and even more exclusive hosting option is Colocation, which in 2005 was used only by a select number of larger customers who chose to house their own servers (and staff) in the host's secure facilities [9, p. 615].

Colocation hosting is similar to dedicated hosting with one key difference: With dedicated hosting, customers rent both the servers and the infrastructure needed to house and maintain them [96]. With colocation, customers own their own hardware but rent server space in a data centre [96]. This allows them to take advantage of improved Internet bandwidth, advanced server cooling and air conditioning systems, and reliable power supplies [96].

2.2.2 Different Hosting Architectures

Apart from the terminology of the offered web hosting services, there is also a separate language for the internal architecture of the systems, which has also undergone numerous changes over the last decades.

The traditional method for constructing a web hosting service, as commonly employed in the industry, also referred to as "sharding", involves the installation of a server, configuration of necessary databases and source code execution environments, and subsequently adding new clients until the capacity of the server is reached, at which point another server is deployed [56].

While being effective, this classic approach is associated with certain limitations: For instance, in the event of a breakdown, recovery can be prolonged, particularly in the absence of real-time system replication, which is cost-intensive [56]. Additionally, the gradual introduction of new technologies, while beneficial in terms of maintaining a high-performance infrastructure, can prove challenging in a rapidly-evolving market such as the internet [56]. This can lead to the need for technical teams to adapt to multiple configurations, increasing the risk of regression or breakdown, and making it difficult for support teams to memorize and keep track of ongoing changes [56]. Furthermore, the presence of specialised teams for different technologies, such as databases, PHP environment, storage systems, and networks, can lead to difficulties in coordination and potential misunderstandings regarding the availability of websites [56]. Lastly, the challenge of isolating customers who consume resources above the average can also have an impact on the performance of other websites hosted on the same server [56].

As a result, multiple different approaches, to cope with these issues have evolved:

N-tier Architecture

High-load websites can benefit from the implementation of n-tier architectures, which allow for the distribution of resources across multiple servers for each software

component [56]. This approach further necessitates the use of a minimum of three servers for the proper functioning of a website, including a server dedicated to the execution of the source code, as well as two separate storage servers for the database and file storage [56], as shown in Figure 2.3. In this architecture, the web server is responsible for receiving requests and executing the website's source code, retrieving data from the file server and database entries provided by the database server [56]. The main requirement for the web server is good CPU resources, which is necessary for executing the source code [56]. File servers, as described in [56], are used to store the files that make up a website and are equipped with specialised hardware to ensure fast data access and durability. Database servers, which are commonly used by websites to dynamically host site data, use a database management system (DBMS) such as MySQL [56]. These servers also require specific hardware, particularly large amounts of RAM, to take advantage of DBMS caching systems and to respond quickly to search queries [56].

As web servers are stateless, they do not store data locally, so it is possible to add more of them and spread the load across multiple machines [56]. This allows websites to be distributed across multiple servers and avoids load balancing across the infrastructure as usage is dynamically distributed across all servers in the farm [56]. In the event of a server failure, other servers in the farm are available to maintain traffic, allowing a wide range of servers in the inventory to be used, provided they have good CPU capabilities [56].

The common use of multi-tier architecture in web applications development, with the inclusion of web servers (e.g. Apache, Microsoft IIS, Nginx), application servers (e.g. Apache Tomcat, Sun Java System Application Server, IBM WebSphere, JBOSS), and database

servers (e.g. Oracle, Microsoft SQL Server, IBM DB2, MySQL, PostgreSQL) is well documented in scientific research [34, p. 467]. In this thesis, however, we will only look at the so-called LAMP stack, that is described in 2.3.1.

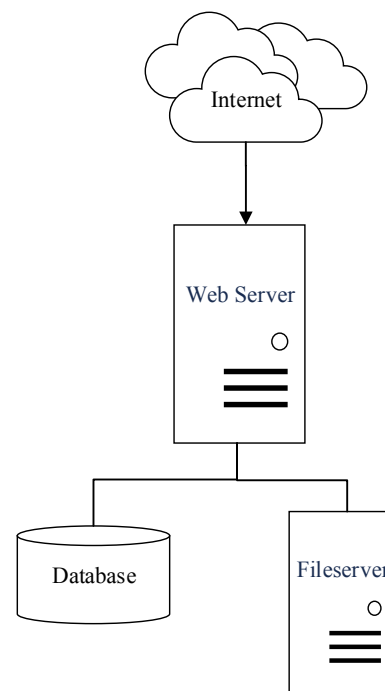


FIGURE 2.3: N-tier architecture.
Source: Own Figure based on [56]

Load balancing

The process of directing queries to the appropriate web server, known as load balancing, is accomplished through the use of specialised entry points [56]. The website traffic is directed to a set of IP addresses dedicated to the web hosting service, which are managed by load balancers that serve as the entry point of the infrastructure [56], see Figure 2.4. These load balancers are effective at handling high volumes of website traffic distributed across multiple web servers through the use of load balancing algorithms [56]. However, the specific requirements of most hosting companies require the use of different sites on a single server and the ability to change the allocation based on various criteria such as the customer's hosting package and the resources required to continuously distribute the load.

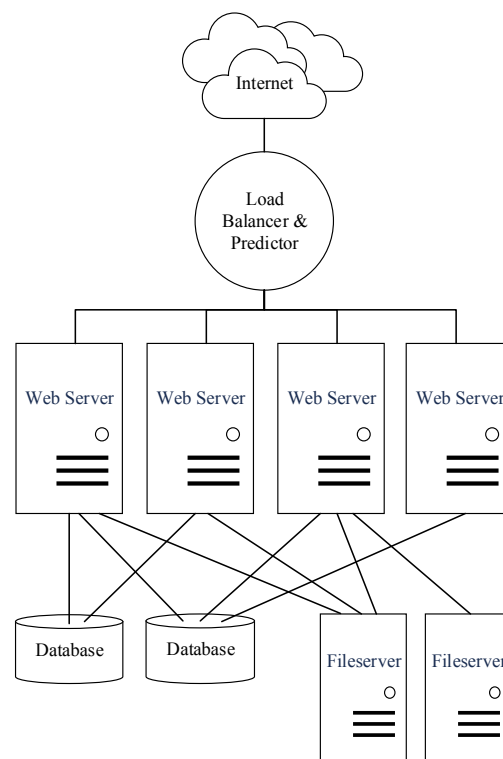


FIGURE 2.4: Load balancing. Source: Own Figure based on [56]

With proper configuration and monitoring, it is possible to dynamically add and remove web servers, also known as elastic computing [47, p. 313]. This approach allows for the automation of resource provisioning, resulting in cost savings while ensuring the ability to handle peak loads. However, there are certain considerations that must be taken into account when implementing this approach [47, p. 313]. The application or website must be designed to run in a distributed manner [47, p. 314]. This includes handling file uploads in a way that ensures all servers have access, which can be achieved by using a clustered file system such as GlusterFS or a centralised object or file storage system such as AWS S3 [47, p. 314]. In addition, the database must be made accessible to all web servers, and session tracking must

also be implemented using a database that is accessible from all servers [47, p. 314]. Fortunately, if a modern framework or content management system (CMS) is used, these aspects are likely to have already been implemented and documented [47, p. 314]. It's important to remember that elastic computing can bring great benefits to the system, but it also requires proper planning and implementation to avoid problems.

Fault domains

The concept of limiting the scope of failures by creating "fault domains", as described in computer science, also exists in other industries such as forestry and construction [56]. The technique of dividing infrastructure into smaller segments and distributing customers across different clusters is also used in the web hosting industry [56]. An example of this approach is the division of OVH's Paris infrastructure into 12 identical clusters, each containing load balancers, web servers and file servers [56], see Figure 2.5.

This approach mitigates the impact of outages, as only a fraction of customers hosted in a given datacentre are affected if a cluster fails [56]. In addition, the database servers are separated from the clusters to allow access to the entire data centre and to enable customers to share their databases between different hosting solutions as required [56].

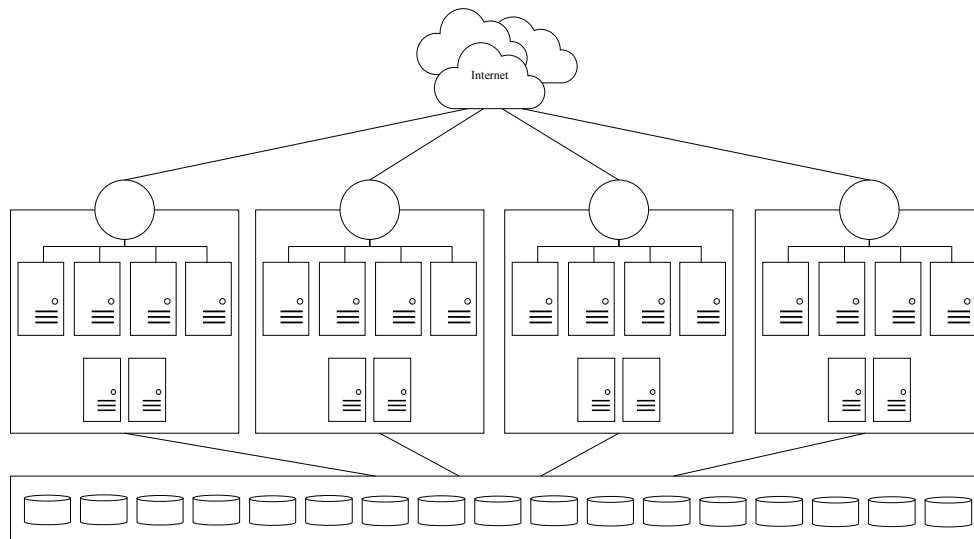


FIGURE 2.5: Fault domains. Source: Own Figure based on [56]

2.2.3 Virtualisation

Virtualisation, a technology that has had a significant impact on the web hosting industry, requires a comprehensive understanding of its underlying concepts and evolution in order to fully understand its impact on the industry.

Definition of Virtualisation

According to Red Hat, the world's leading provider of enterprise open source solutions, now part of IBM, "Virtualisation is technology that allows you to create multiple simulated environments or dedicated resources from a single, physical hardware system." [89] The concept of virtualisation provides abstraction on top of physical resources, allowing for the efficient utilisation of hardware resources [64, p. 2].

There are various levels of abstraction in virtualisation, with the two most prevalent being virtual machine (VM)-based and container-based [64, p. 2]. VM-based virtualisation creates a virtualised version of the entire hardware stack, including the operating system, while container-based virtualisation abstracts the operating system layer to create isolated environments for applications [64, p. 3]. Other techniques, such as unikernels, which are lightweight, single-purpose virtual machines, are also being explored [64, p. 3].

This thesis is focused on the analysis of VM-based virtualisation techniques. The VM-based approach utilises virtualisation at the operating system level, providing an abstraction of the underlying physical resources through the use of virtual devices, such as virtual disks, virtual CPUs, and virtual network interfaces [64, p. 3]. This approach allows for the creation of virtualised environments that emulate the functionality of a physical machine, enabling the execution of multiple operating systems on a single physical host [64, p. 3].

The History of Virtualisation

The advent of virtualisation technologies also allowed an increased differentiation of the hosting services offered. The history of virtualisation goes back to the mainframe days in the late 1960s and early 1970s. Computers were big and expensive machines. Time-sharing was the most used way to increase the capacity utilisation of computing resources by sharing usage among a large group of users [16]. It started with the development of IBM's CP-40 mainframe system [64, p. 1]. This system evolved into the CP-67, which used partition technology to run multiple applications simultaneously [64, p. 1]. The advent of UNIX further advanced the ability to run multiple programs on x86 hardware [64, p. 1]. However, portability remained an issue [64, p. 2]. In the early 1990s, Sun Microsystems introduced Java, which allowed for the "write once run anywhere" paradigm by introducing intermediary code, or bytecode, which could be executed on a Java runtime environment across different hardware architectures [64, p. 2]. This marked the emergence of process-level virtualisation. In the late 1990s, VMware introduced its own virtualisation model, which virtualised actual hardware components such as the CPU, memory, and disks, allowing for the running of operating systems themselves as guests on top of the VMware software, or hypervisor [64, p. 2].

Virtualisation Software

VMware is the foremost provider of virtualization services and offers a comprehensive virtualisation suite, known as "VMware SDDC" (Software-Defined Data Center), that includes a variety of solutions for virtualizing servers, storage, and networking [59, p. 177]. The suite includes the popular "vSphere ESXi" hypervisor, which allows for the creation and management of virtual machines, as well as other products such as Horizon for virtual desktop infrastructure and "NSX" for software-defined networking [59, p. 177].

On Windows Servers the virtualisation software "Hyper-V" is used, a hypervisor technology developed by Microsoft for use on Windows-based operating systems [59, p. 174]. It allows for the creation and management of virtual machines, enabling the efficient utilisation of hardware resources and the ability to run multiple operating systems on a single physical server [59, p. 174]. Additionally, Hyper-V supports a wide range of guest operating systems, including Windows and Linux [59, p. 174].

Virtualised Hosting

This type of server virtualisation also allows hosting companies partition a single physical server into multiple virtual servers, each with its own operating system, software, and resources. This enables multiple applications and services to be run on a single physical server, which can be more efficient and cost-effective than running them on separate servers. But these simulated environments must be controlled and managed, which is done by the so-called *hypervisor* [39, p. 20], see Figure 2.6.

This type of software is run directly on the hardware and "allows to split one system into multiple separate, distinct and secure environments known as "virtual machines (VMs)"" [89]. These virtual machines rely on the hypervisor's ability to isolate the machine's resources from the hardware and allocate them appropriately [89].

The physical hardware that is equipped with a hypervisor is referred to as the *host*, while the multiple VMs that utilise its resources are called *guests* [89]. These guests perceive computing resources, such as CPU, memory, and storage, as a pool of resources that can be easily relocated [89]. Operators can control virtual instances of CPU, memory, storage, and other resources, ensuring that guests receive the resources they require when needed [89].

Virtual Private Server

Virtualisation enables hosting providers to divide the resources of a single physical server into multiple virtual servers, each of which can be configured and sold as a separate hosting account. These virtualised environments are offered as "virtual private server" (VPS) and are configured to emulate a dedicated server with its own set of resources [31, p. 224]. This is achieved by utilizing virtualisation software that

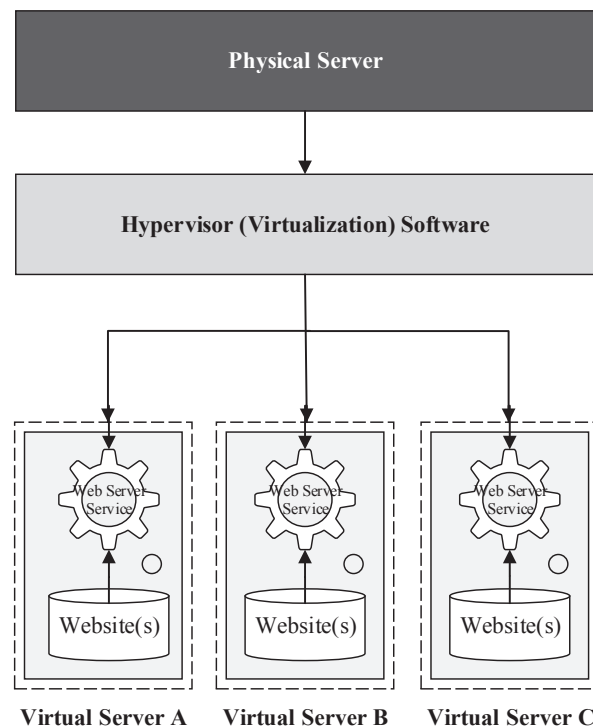


FIGURE 2.6: Virtualised Hosting Concept. Source: Own Figure based on [51]

runs on a physical server and divides it into multiple virtual instances, each of which is allocated a specific amount of resources [31, p. 224]. The hypervisor is responsible for the allocation, release, and oversight of the resources of guest operating systems, also known as virtual machines [39, p. 20].

This technique is distinct from “shared hosting”, described in 2.2.1, in that the virtual servers are logically separated from one another [39, p. 19]. Therefore each VPS is typically self-contained [31, p. 224], which means, that, in theory, each VPS is only able to access the resources allocated to it; however, in practice, many are configured to allow some overflow, meaning that resources from other VPSs on the same dedicated box can be borrowed [31, p. 224] as confirmed in the interviews in Chapter 5. This implies that, although in theory a VPS is guaranteed to have 2GB of RAM, if another VPS on the same server experiences high utilisation, it may take some of the memory and processor time from the other VPSs [31, p. 224].

Benefits of Virtualisation

The importance of virtualisation technology is illustrated by the following quote: “This model represented a major breakthrough in computer technology: the cost of providing computing capability dropped considerably and it became possible for organisations, and even individuals, to use a computer without actually owning one. Similar reasons are driving virtualisation for industry standard computing today: the

capacity in a single server is so large that it is almost impossible for most workloads to effectively use it. The best way to improve resource utilization, and at the same time simplify data center management, is through virtualization.” [27]

Since hosting providers have to hold enormous hardware resources and provide them to customers on demand, the need for more efficient use of resources is also imperative in order to be able to offer competitive prices. Hence, the “increasing adoption of virtualization technology in the web hosting industry has led to many advancements, making it an integral part of the current web hosting ecosystem: It is commonplace to use virtualized server environments for hosting web applications, as well as for general distributed computing systems that host various web applications.” [18, p. 227] Optimizing allocated resources can reduce both operating costs and energy consumption [18, p. 227]. By hosting in a virtualized server environment, researchers could demonstrate that the CPU capacity allocated to the website can be optimized as the workload changes [18, p. 227]. In doing so, control structure for a processor sharing system where the allocated portion of CPU capacity can be set at fixed time intervals minimized the allocated capacity while maintaining the SLA [18, p. 227].

Virtualisation enables the consolidation of workloads onto fewer physical platforms, resulting in increased resource utilisation and hardware optimisation. Initially a technology that enabled computing enthusiasts to run multiple operating systems concurrently on a single PC, hardware virtualisation has evolved into an indispensable tool for gaining improved business efficiencies. As noted by Bose [14] and Bazargan [28], the use of it can lead to various benefits, including:

- Consolidation of multiple operating environments onto one server, which results in a significant reduction of capital expenditure by utilizing a fraction of the computing resources of a server. Additionally, the reduction of physical servers also leads to lower power, cooling, and space costs as well as lower operating expenses [14]. As noted by Bazargan [28], the primary goal of virtualisation is to combine and unify workloads on fewer physical platforms that can sustain their demand for computing resources, such as CPU, memory, and I/O. As a result virtualisation can dynamically allocate resources to a single logical or physical partition at a given runtime.
- The ability to migrate a virtual machine to a new physical server while applications on the VM continue to be in use, thus allowing for maintenance operations on the original server [14]. It also enables a rich set of load balancing and resource management features [14].
- Virtualisation provides a high level of reliability by maintaining functionality and operation availability through the isolation of virtual machines. The achievement of high availability (HA) by allowing a VM to be restarted on a new server if the server running the VM fails [14]. According to Bazargan [28],

a system fault on one VM or its partition will have no effect on other partitions that are running on the same physical platform.

- In terms of security, virtualisation also offers added value through the encapsulation and isolation of virtual machines' operating systems. As stated by Bazargan [28], if a certain partition or service on a VM is compromised, this stops the compromise from being extended to other partitions that are existent on the same virtualisation platform.

2.2.4 Cloud Computing

Virtualisation is also considered as the fundamental enabling technology behind cloud computing [14]. Not only can a single physical server host multiple virtual servers, each with its own set of resources, but today it is also possible to bundle multiple physical servers in virtual clouds. Within these clouds, resources can be dynamically transferred from one host to another based on the state of system load and the current requirements of individual users or applications [89]. This means, virtualisation enables hosting providers to create and manage their virtual servers in the cloud, which can be customized and scaled up or down as needed to meet the changing needs of the applications.

As the following quote indicates, the differences between the concept of cloud computing and virtualisation lie in the degree of abstraction: "It's easy to confuse virtualisation and cloud, particularly because they both revolve around creating useful environments from abstract resources. However, virtualisation is a technology that allows you to create multiple simulated environments or dedicated resources from a single, physical hardware system, and clouds are IT environments that abstract, pool, and share scalable resources across a network. To put it simply, virtualisation is a technology, where cloud is an environment." [121]

Cloud Hosting and Cloud Services

The fundamental concept behind cloud computing is the aggregation of multiple computers to achieve faster and more reliable performance [31, p. 26]. In the context of web hosting, this translates to faster delivery of websites and a reduced risk of slowdowns. When a particular website receives a high volume of traffic, the load is distributed evenly across multiple servers in the cloud [31, p. 26]. This means, in cloud hosting, multiple physical servers are connected together to form a cloud-based infrastructure that can be used to host websites and web-based applications.

It has been acknowledged in literature [15], that traditional hosting methods present certain drawbacks, that cloud hosting fixes. One of the main issues is that the resources of a single server are shared among multiple websites, which can result in decreased performance during spikes in traffic to those other sites [48, p. 234]. Additionally, security breaches and other performance issues on other sites hosted

on the same server can also negatively impact the performance of an individual website [48, p. 234]. Furthermore, traditional hosting presents a single point of failure, meaning that if the server itself experiences technical problems, it can affect all websites hosted on that server [48, p. 234]. In contrast to this, cloud hosting offers a level of scalability that traditional hosting methods may not be able to provide [48, p. 234]. The load is distributed across a cluster of multiple servers, with the information and applications contained on those servers mirrored across the entire cluster [48, p. 234]. This redundancy ensures that if an individual server goes down, there is no loss of information or downtime [48, p. 234]. This increased elasticity and resilience makes cloud hosting a more reliable option, as problems with one website or application are less likely to have an impact on bandwidth or performance [48, p. 234].

It is also important to note, that billing models of cloud hosting offers are fundamentally different from those of traditional web hosting packages. Rather than incurring costs for a fixed amount of space on a single server, with cloud hosting, the user pays for the resources they actually utilise [48, p. 234], what makes the estimation of possible costs difficult, but is not within the scope of this thesis.

Infrastructure as a Service (IaaS)

The emergence of cloud computing has significantly transformed the paradigm of web application hosting [77, p. 99]. The process of procuring and provisioning new infrastructure, which previously entailed a substantial amount of time and resources, can now be accomplished with minimal effort through the utilisation of cloud providers [77, p. 99]. These providers offer a range of virtual machines, referred to as infrastructure as a service (IaaS), and managed software solutions such as databases, on a pay-per-usage model, thus enabling users to only pay for the resources they utilise [77, p. 99]. Cloud hosting companies typically provide so called Infrastructure-as-a-Service (IaaS) [48, p. 234]. Which are, according to Gartner, “a standardized, highly automated offering in which computing resources owned by a service provider, complemented by storage and networking capabilities, are offered to customers on demand. Resources are scalable and elastic in near real time and metered by use. Self-service interfaces, including an Application Programming Interface (API) and a graphical user interface (GUI), are exposed directly to customers. Resources may be single-tenant or multi-tenant, and are hosted by the service provider or on-premises in a customer’s data center.” [97] This flexibility has enabled engineers to rapidly create and dissociate testing environments as needed, according to De Jonghe [77, p. 99].

The diagram 2.7 illustrates the progression from traditional on-premise IT infrastructure to colocation and hosting to Infrastructure-as-a-Service (IaaS), with expanding service offerings under the purview of the provider, as well as the distinction from Platform-as-a-Service (Paas) and Software-as-a-Service (Saas).

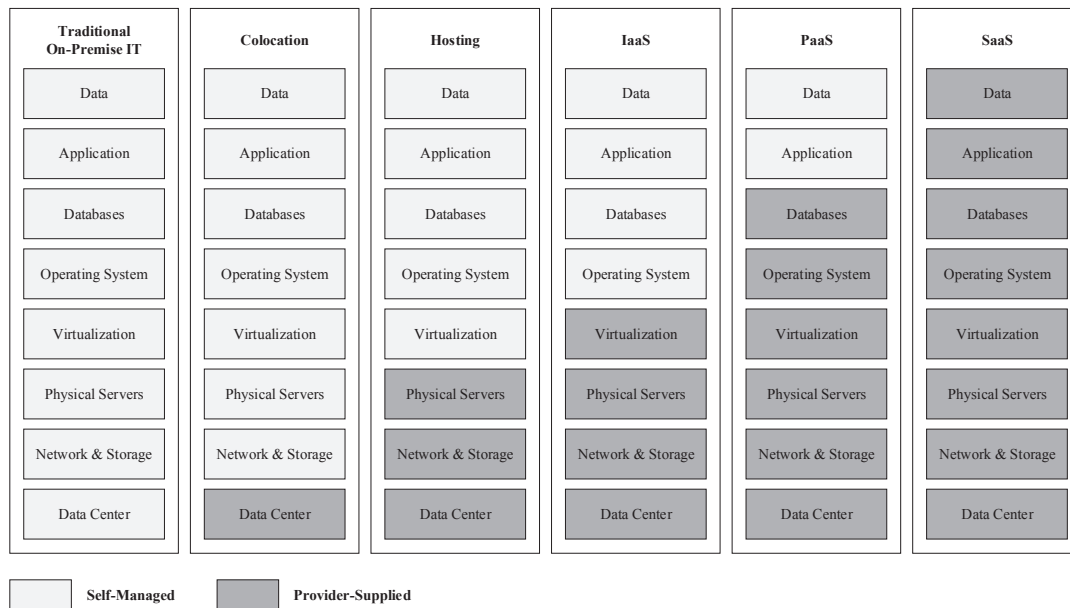


FIGURE 2.7: On-Premise, Colocation, Hosting, IaaS, SaaS and PaaS compared. Source: Own Figure based on [53]

2.2.5 Content Delivery Networks

A Content Delivery Network (CDN) is employed to host static media files such as images, .css and .js files, audio and video files on a network of servers located in various geographical locations [40, p. 52]. The files are then served to requests from a specific server, depending on the location of the request [40, p. 52]. This offers several benefits such as: The caching of static content in memory, enabling faster delivery of files from cache as opposed to loading them from disk and sending them to the browser [40, p. 52]. The distribution of files across different locations, allowing for faster delivery of content from the nearest location to the requested location, thus reducing response time [40, p. 52]. The ability to utilise subdomains, either provided by the CDN or using the main domain's DNS settings, enabling browsers to send more parallel requests for the same content loading from different domains, resulting in faster page loading [40, p. 52]. The ability to offload the load on the server by hosting the static content on a separate CDN server, which reduces the number of requests for dynamic content and improves the performance of the application [40, p. 52].

2.2.6 Managed Services

Like mentioned in the beginning, the hosting industry is characterized by a dynamic and ongoing evolution. The recurrent phenomenon of companies attempting to differentiate themselves from their competitors through the implementation of unique services and offerings is obvious. It also can be observed, that when such novel services are positively received in the market, competing companies often quickly adopt similar strategies. One example of this phenomenon is the offer of *managed*

hosting, which has attracted a lot of attention in recent years. In the hosting industry, managed services refer to an additional level of support and assistance that goes beyond the typical offerings of a VPS or dedicated server [31, p. 226]. Managed hosting, which generally incurs additional costs, includes additional services such as the installation of the operating system, server software, and security software by a system administrator. Some packages may also include automatic updates of server software [39, p. 22]. In contrast, in “unmanaged” hosting environments, the user is responsible for complete management of the server, including the installation of the operating system and website software [39, p. 21].

It is important to note that “Managed Hosting” as well as “Managed WordPress Hosting” are not trademarked phrases. Consequently, the definition of this term may vary from provider to provider [90] and the price and included services of managed hosting vary greatly [39, p. 22]. [90] Each company has its own version of managed services, with varying levels of support and assistance [31, p. 226]. At its core, managed services allow the hosting company to take over many of the technical, backend aspects of hosting, providing a user experience similar to that of a shared server, but with the speed and power benefits of a VPS or dedicated server [31, p. 226].

Regarding “Managed WordPress Hosting”, automated daily backups and free Secure Sockets Layer (SSL) certificates are expected to be included in any offer [90]. Some providers may offer ordinary auto-updates, while others may restrict access to system files or installable plugins [90]. But the interpretation of this term is generally left to the provider, which is why the term “managed” is not very meaningful [90]. Some hosting providers try to place even more emphasis on the service component. For an example, they offer a seamless transfer of websites to their system and may also include individual performance optimisation [90]. In the event of malware attacks, these providers promise rapid assistance, including analysis and correction of performance issues, as well as consulting and load tests for larger campaigns such as TV appearances [90].

2.3 Software Concepts

Computer system resource management has attracted much attention in recent years because poorly managed resources can severely degrade computer system performance and due to high operation costs [18, p. 226]. Therefore, it is important to understand the common software concepts used in web hosting services.

2.3.1 Web Servers

The LAMP software package, which stands for Linux, Apache, MySQL and PHP, is widely utilised as the underlying technology for a significant proportion of websites currently being offered in the World Wide Web [36, p. 13]. This software combination had become a de facto standard in the realm of web development due to its stability,

flexibility and ease of deployment. But the "A" is not only standing for Apache exclusively, because NGINX and Apache are the two most widely utilised HTTP servers [40, p. 41].

Apache

Apache, one of the pioneering web servers on the World Wide Web, has retained its position as the most widely employed web server in the world today [36, p. 13]. Despite its longevity, the Apache web server continues to receive active maintenance and development efforts, thereby ensuring its continued relevance and effectiveness in the field of web server technology [36, p. 13]. Apache is widely recognized as a popular choice among administrators for its flexibility, broad support, versatility, and modules for various interpreted languages, such as PHP [40, p. 42]. Given its ability to process a vast number of interpreted languages, it does not require communication with other software to fulfill requests [40, p. 42].

Apache offers multiple processing options such as prefork (processes spawned across threads), worker (threads spawned across processes), and event-driven (dedicated threads for keep-alive connections and separate threads for active connections) which offers greater flexibility [40, p. 42]. However, it is important to note that each request is processed by a single thread or process, thus Apache's usage of system resources can be high [40, p. 42]. Therefore, in high-traffic applications, this can lead to a decrease in performance as it does not provide optimal support for concurrent processing [40, p. 42].

NGINX

NGINX was designed to address the issue of concurrency in high-traffic applications by providing an asynchronous, event-driven, and non-blocking approach to request handling [40, p. 42]. As requests are processed asynchronously, NGINX does not block resources while waiting for a request to be completed [40, p. 42]. Instead, NGINX creates worker processes, each of which can handle a substantial number of connections, enabling a limited number of processes to efficiently handle high traffic [40, p. 42].

NGINX is a software suite that has evolved beyond its initial conception as a basic web server package and has gained widespread recognition for its advanced features such as reverse proxy and load balancing capabilities [47, p. 55]. Its inception was motivated by the need to address the C10k problem, which pertains to the ability to handle 10,000 concurrent connections [47, p. 55]. The event-driven architecture employed by NGINX sets it apart from Apache, which has also incorporated event-driven processing in its latest version 2.4 [47, p. 55]. However, there are several distinctions that render NGINX more versatile in comparison to Apache [47, p. 55].

It should be noted that NGINX does not have built-in support for interpreted languages and instead relies on external resources for this purpose [40, p. 42]. This design decision allows the processing to occur outside of NGINX, with NGINX only responsible for handling connections and requests [40, p. 42]. While NGINX is often considered to be faster than Apache, particularly in the case of serving static content, it should be acknowledged that in high-performance servers, it is not always Apache that is the bottleneck, but rather the PHP interpreter [40, p. 42].

PHP, SAPI, CGI and FastCGI

PHP is a widely adopted scripting language for web development [36, p. 14]. It is characterized by its ease of use and rapid learning curve, which has contributed to its popularity among developers [36, p. 14]. A significant number of popular blogs and content management systems, such as WordPress, Drupal, and Joomla, have been built using PHP [36, p. 14]. PHP is an interpreted language, which means that PHP scripts are in plain-text and can be easily read by humans [36, p. 14]. When a browser requests a PHP page, the web server must pass the request to the PHP engine to interpret and execute the code, and then return the results back to the Apache web server [36, p. 14]. The web server then displays the results in HTML to the user's browser [36, p. 14].

The communication between the web server and PHP engine is facilitated by the PHP Server API (SAPI) or Apache Handler [36, p. 15]. The Common Gateway Interface (CGI) is the legacy SAPI for Apache and a neutral protocol that enables web servers to connect with a variety of language interpreters, including PHP, Perl, and Python [36, p. 20]. This means, that instead of the PHP engine running inside each child process, Apache uses the CGI method to invoke an external program, such as `/usr/bin/php-cgi`, to process PHP scripts [36, p. 20]. The web server then receives the output from the CGI program and displays it to the browser [36, p. 20]. This approach allows for greater flexibility in terms of the language interpreters that can be used with the Apache web server [36, p. 20].

The primary disadvantage of the CGI SAPI is its high CPU utilisation and consequent decrease in performance relative to other methods [36, p. 20]. The Apache web server invokes the `php-cgi` process for each client request and closes it after the request has been processed [36, p. 20]. This approach entails an overhead of starting a new CGI process for each client request and as a result, it is not possible to use an "Opcode Cache" to enhance performance [36, p. 20]. This is due to the non-persistent nature of the CGI processes, which are opened and closed by Apache after the completion of each request [36, p. 20].

FastCGI is an advancement of the CGI protocol [36, p. 20]. The key distinction between FastCGI and CGI is that FastCGI is persistent, meaning that the FastCGI processes are kept alive by a process manager (pm) and can be reused for subsequent

client requests [36, p. 20]. This results in a reduction of CPU overhead associated with spawning a new CGI process for each client request [36, p. 20]. FastCGI today is the preferred method for running PHP with Apache, in which the PHP processes are kept separate from the Apache processes [36, p. 20]. When a client requests a PHP script, Apache routes the request to the PHP engine through the FastCGI protocol [36, p. 20]. On NGINX, all PHP files are processed via the FastCGI interface to PHP-FPM [47, p. 96].

Opcache as Performance Enhancement Extension

As an interpreted language, the PHP scripts must be parsed, compiled into "opcodes", and then executed by the PHP engine [36, p. 51]. However, this process can be time-consuming, when a web browser requests a PHP page or script [36, p. 51]. To address this issue, the "Opcode Cache" was introduced as a performance enhancement extension for PHP scripts [36, p. 51]. An opcode cache stores the parsed and compiled version of a PHP script in memory, allowing subsequent requests for the same script to be executed immediately without the need for parsing and compilation [36, p. 51].

Data Base Management Systems

Databases serve a crucial function in dynamic websites as they store and manage all incoming and outgoing data [40, p. 73]. Therefore, an ill-designed and unoptimised database for a PHP application can have a significant impact on its performance [40, p. 73]. As mentioned above, the "M" component of the LAMP stack refers to the MySQL server, a widely adopted Relational Database Management System (RDBMS) for the web and is open-source with a free community version [40, p. 73]. It offers a comprehensive set of features that are typically found in enterprise-level databases [40, p. 73]. However, it is important to note that the default settings provided with the MySQL installation may not be optimal for performance and can be fine-tuned to improve performance [40, p. 73]. Prior to its acquisition by Sun Microsystems and later Oracle Corporation, MySQL was considered the most prevalent open-source option in this category [36, p. 14].

It is worth noting that some individuals assert that the "M" in MySQL may also refer to MariaDB, which is considered as a "fork" of the original MySQL [36, p. 13]. In recent years, this alternative option, MariaDB, has gained popularity as a binary-compatible drop-in replacement for MySQL [36, p. 14].

Key-Value Cache Stores

Key-Value Cache Stores like "Redis" or "Memcached" helps in improving the performance of a website by storing data (strings or objects) in memory, thus reducing the communication with external resources such as databases or APIs [40, p. 95]. Redis is an open-source in-memory key-value data store that is commonly utilised

for database caching. As stated on the official Redis website [109], Redis supports various data structures including strings, hashes, lists, sets, and sorted lists, as well as features such as replication and transactions [40, p. 95]. As stated on the official website of Memcached [120], it is a free, open-source, high-performance, and distributed memory object caching system. Memcached is an in-memory key-value store that can store datasets from a database or API calls [40, p. 95].

2.3.2 Control Panels

Web hosting control panels provide a graphical user interface that facilitates the management of server-level commands, making it more accessible for non-technical users [99]. Many web hosting providers include control panels as part of their hosting packages, simplifying the process of configuring and monitoring essential web server functionality [99]. Control panels typically provide a range of features and tools that enable users to perform various tasks related to hosting their websites, such as:

1. Setting up and configuring their hosting accounts and websites
2. Managing and organizing their files and data
3. Setting up and managing email accounts and email forwarding
4. Monitoring website performance and resource usage
5. Installing and configuring web applications and software
6. Backing up and restoring data
7. Monitoring and managing security features [99]

Web hosting control panels are typically provided by hosting providers as a service to their customers, and are accessed through a web browser using a unique login and password [6, p. 381]. There are various types of web hosting control panels available, each with its own set of features and tools. Two of the most popular control panels are "cPanel" and "Plesk" [6, p. 381]. Overall, web hosting control panels provide a convenient and user-friendly way for users to manage and maintain their hosting accounts and websites, and are an essential tool for anyone hosting a website or web-based application.

Although the control panel provided could have an impact on the selection of a hosting provider, these have no direct influence on the measurable performance. For this reason, they are not the subject of consideration in this paper.

2.3.3 Content Management Systems

A content management system (CMS) is a software application that facilitates the management and organisation of digital content for web publishing [6, p. xxvii]. The functionality of a CMS can be tailored to the specific needs of a community by

including or excluding certain features [6, p. xxvii]. Common features of a CMS include file management, photo galleries, private messaging, discussion forums, articles, polls, and many others [6, p. xxvii]. These systems are widely used by online newspapers, magazines, and other news sources to manage their web presence [6, p. xxvii].

Covering nearly 30 percent of all websites, WordPress is certainly the Content Management System (CMS) of choice by many. Although it came from a blogging background, WordPress is a very powerful CMS for all content types and powers some of the world's busiest websites [47, p. 91]. Therefore the WordPress Platform is of particular interest for this work.

2.4 Web Performance Metrics

The increasing popularity of online services has led to a significant interest in improving both, server hardware and software [41, p. 122]. The rapid growth of the web performance optimisation industry in recent years highlights the increasing importance of speed and user experience in the digital realm [29, p. 3]. The importance of performance in software engineering is a widely acknowledged aspect of the field. One of the most salient reasons for this is that it directly impacts the user experience [25, p. 1]. This means, if an application functions at a faster rate, that will provide a more satisfactory experience for the end user. Additionally, performance is also often considered a key metric in determining the overall quality and usability of an application [25, p. 1]. This is not simply a result of the psychological need for speed in our fast-paced and connected world, but rather a requirement driven by empirical evidence [29, p. 3]. Studies have shown that faster websites lead to better user engagement, retention, and conversions, emphasizing the significance of speed as a key feature for online businesses [29, p. 3]. As such, it is a crucial consideration in the development and maintenance of software systems.

The loading speed of websites, often referred to as "Page Speed", reflects the website's performance and has a significant influence on user experience and satisfaction [35]. It is sometimes being interchanged with the term "performance". According to Barker, performance is a metric that quantifies the operational efficiency of an application and a multifaceted aspect of quality [25, p. 1]. But as the following quote shows, it not an easy task, because: "Optimizing the interaction among all the different layers is not unlike solving a family of equations, each dependent on the others, but nonetheless yielding many possible solutions." [29, p. 165] and, that there "is no one fixed set of recommendations or best practices, and the individual components continue to evolve." [29, p. 165] Web performance is essentially best described as the time that the content takes to be delivered to the end user, including network latency and browser render time [25, p. 1].

2.5 Network Performance

Backbones, that constitute the core data paths of the Internet have the capacity to transmit hundreds of terabits per second [29, p. 10]. However, the available capacity at the edges of the network, which can vary significantly depending on the technology being utilised (e.g. dial-up, DSL, cable, wireless technologies, fiber-to-the-home, local router performance), is significantly lower and ultimately determines the available bandwidth for the user through the identification of the lowest capacity link between the client and the destination server [29, p. 10]. Most important, determining the location of the bandwidth bottleneck for a specific user is often a complex yet crucial task [29, p. 10].

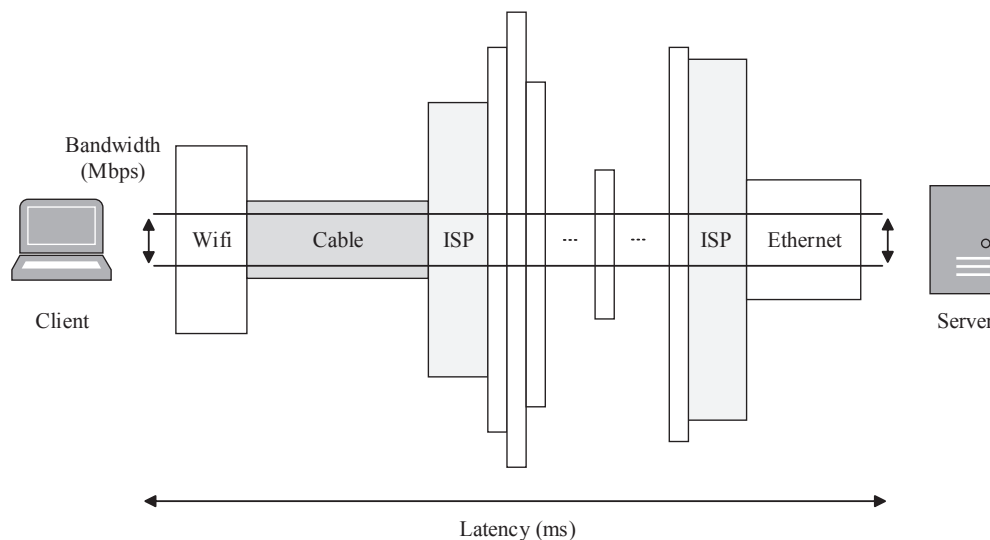


FIGURE 2.8: Latency and bandwidth. Source: Own Figure based on [29, Figure 1-1]

While Latency is “the time from the source sending a packet to the destination receiving it” [29, p. 3], bandwidth is “the maximum throughput of a logical or physical communication path” [29, p. 3]. To be more precise, latency refers to the time it takes for a message or packet to travel from its point of origin to its destination. However, this simple definition often obscures the complexity of the underlying systems and components that contribute to latency. To truly understand latency, it is important to analyze the four components that contribute to the overall time it takes for a message to be delivered: [29, p. 4]

- “Propagation delay” is the “amount of time required for a message to travel from the sender to receiver, which is a function of distance over speed with which the signal propagates.” [29, p. 5]
- “Transmission delay” is the “amount of time required to push all the packet’s bits into the link, which is a function of the packet’s length and data rate of the link.” [29, p. 5]

- “Processing delay” is the “amount of time required to process the packet header, check for bit-level errors, and determine the packet’s destination.” [29, p. 5]
- “Queuing delay” is the “amount of time the incoming packet is waiting in the queue until it can be processed.” [29, p. 5]

The resulting sum of all these delays is the total latency between the client and the server [29, p. 5].

Chapter 3

Research Design

Web science is an interdisciplinary field. It draws knowledge from multiple disciplines such as sociology, anthropology, psychology, economics, and computer science. Web science combines research from these different fields like quantitative, qualitative and mixed methods research, including techniques from the social sciences and computer science to understand the complex and multiple impacts of the Web on society and vice versa [117]. This interdisciplinarity also consists in the fact that, unlike other scientific disciplines, there is not only one theory or one method with which scientific work is done [54]. Therefore, a combination of different scientific methods are used in this thesis.

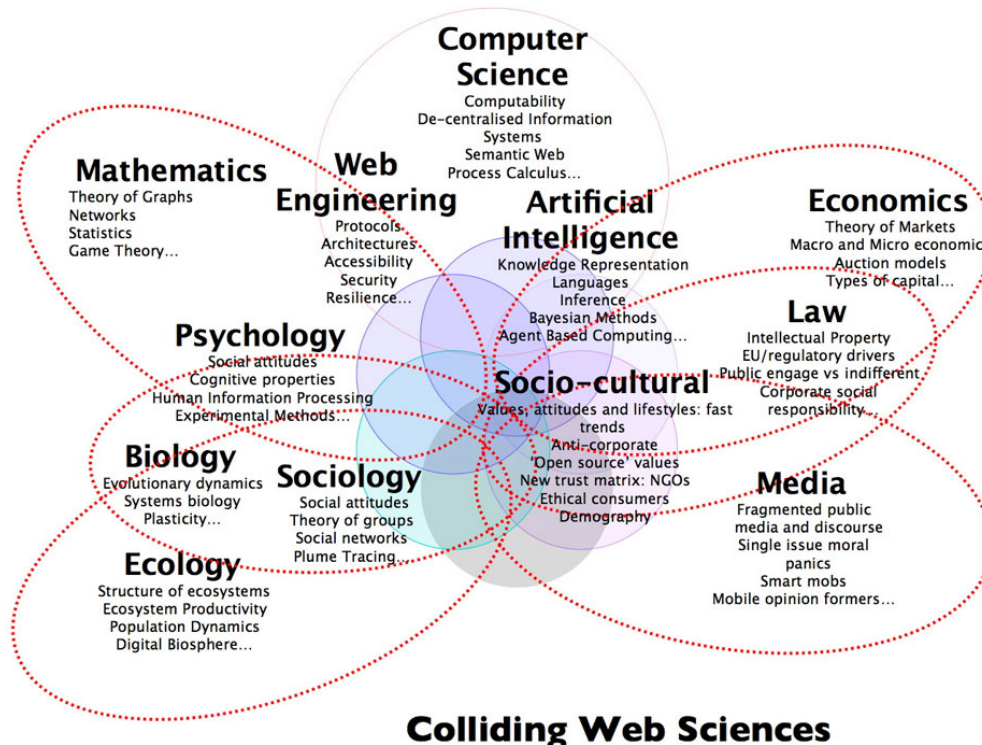


FIGURE 3.1: The Web Science 'butterfly'. Source: [26]

The complexity of the problem, addressed in this thesis, also stems from the various factors that can affect web hosting performance, such as hardware resources, network infrastructure, software configurations and user perception and necessitates the use of a multi-faceted research approach.

3.1 Research Goals

The primary objective of this master thesis is to design and propose a methodology for evaluating the performance of web hosting solutions in the context of hardware virtualisation. The goal of this research is to provide a systematic approach for benchmarking web hosting solutions, which will facilitate the comparison of offerings from various providers and assist organisations in selecting the most appropriate web hosting service. By providing a systematic approach for benchmarking, the thesis aims to provide a tangible solution that can be used to evaluate and compare web hosting offerings. The research questions are listed in more detail in *Chapter 2*.

There is a substantial body of research related to benchmarking in the field of software engineering, however, it is important to note that a significant proportion of this research does not specifically focus on managed hosting or shared hosting environments. This highlights a gap in the current research in this area, as managed hosting and shared hosting environments have unique characteristics that may require specialised benchmarking techniques and methods. Therefore, it would be valuable to conduct further research in this area to gain a better understanding of how to accurately evaluate the performance of managed and shared hosting environments.

3.2 Research Methods

Quantitative research methods, such as statistical analysis and surveys, are the dominant research framework in the social science and typically used to collect and analyze numerical data [33]. They are useful for identifying patterns and trends in data, but may not provide in-depth insights into why these patterns exist. But, quantitative research relies on data that are observed or measured to examine questions about a sample group of people [44]. In the context of this thesis, the focus was not to examine a specific target group, user group or even group of web hosting providers, but to find a new approach to benchmarking. Therefore, no quantitative methods were used.

Qualitative research is particularly useful for understanding the nature of phenomena and answering questions about why something is or is not observed [61]. This type of research is used in the study of complex interventions and to focus on ways to improve them [61]. Qualitative research methods, such as document study, observations, semi-structured interviews, and focus groups, are used to collect and analyze non-numerical data [61]. The data is then analyzed by transcribing field notes and

audio recordings into protocols and transcripts, and using qualitative data management software for coding [61]. They therefore provide in-depth insights into people's experiences, beliefs, and attitudes, but may not be as useful for identifying patterns and trends in data.

Given the multi-faceted nature of this problem, it was determined that a one-dimensional research design would not be sufficient to provide an accurate and comprehensive assessment of the complex field of web hosting performance benchmarking. First, it was necessary to understand how modern web hosting systems are structured so that the existing limitations and peculiarities in performance measurement could be adequately captured. As a result, a multi-layered approach was adopted to effectively capture the various aspects of both, the relevant findings and current state of academic research and the professional experience and best practices of experts. Subsequently, existing methods for measuring performance could be recorded and then examined for their suitability in the extremely limited environment of shared hosting.

Three different appropriate research methods and models were selected to adequately answer the primary research questions:

3.2.1 Literature review

The literature review in this thesis is a combination of research from two angles:

In the first part of the literature review, published academic research is analyzed for important references related to hosting architectures and existing approaches to benchmarking. The focus is on possible limitations and findings, both from theoretical and practical research related to the evaluation of performance with respect to web hosts.

The second part of the literature review is a theoretical examination of the architecture and design of web hosting environments, drawing on relevant literature aimed at server administrators and DevOps professionals. The focus is on current best practices, commonly used software and configurations, and the technical requirements, constraints, and opportunities associated with these environments.

Research on Academic Publications

The procedure for conducting the literature review with the goal to develop an approach of a web hosting benchmark in times of virtualisation can be broken down into the following steps:

1. **Identifying key terms and concepts related to the topic:** This step aims to find the main terms and concepts that are relevant to the topic of web hosting performance measurement in virtualised environments. Some examples of these terms could include web hosting, performance measurement, virtualisation,

shared and managed environments, server administration, and DevOps. It's important to have a good understanding of these terms and concepts in order to effectively search for and evaluate relevant literature.

2. **Searching for relevant literature:** In this step, the key terms and concepts identified in step 1 were used to conduct a search for relevant literature. A variety of databases that were used for this purpose, such as Google Scholar, JSTOR, IEEE Xplore, and ACM Digital Library. When searching for literature, it was important to use a variety of search terms and to use both, broad and specific terms.
3. **Reading and evaluating the literature:** After identifying a set of relevant literature, the next step was to read and evaluate the literature. This includes reading the abstracts of articles and the introductions of books to determine whether they are relevant to the topic. It was also important to evaluate the literature based on its credibility and reliability, looking at the authors' qualifications, the journal or publisher, and the methodology used in the research.
4. **Organizing selected literature:** After identifying the most relevant literature, the next step was to organise the literature into categories based on the main themes that emerge from the research. This included categories such as web hosting, web development, performance measurement techniques, virtualisation technologies, and benchmark development. Organizing the literature in this way made it easier to identify the main findings from the research.
5. **Writing the literature review:** The literature review is an important part of this thesis, that summarises the main findings from the literature and discusses how they relate to the goal of the thesis. In this step, the literature review is written, drawing on the selected and organised literature. The literature review should provide a comprehensive overview of the current state of research on the topic and highlighting the main findings.
6. **Identify gaps in the literature and areas for further research:** In this final step, gaps in the literature and areas that require further research were identified. This includes areas where there is a lack of research, where research is inconclusive, or where there is a need for more in-depth research. Identifying these gaps in the literature help to guide the direction of the research for the remainder of the thesis.

Research on professional literature

The second part of the literature review is a theoretical examination of the architecture and design of web hosting environments, drawing on relevant literature aimed at server administrators and DevOps professionals. The focus is on current best practices, commonly used software and configurations, and the technical requirements, constraints, and opportunities for benchmarking, associated with these environments.

The procedure for conducting this literature review can be broken down into the following steps:

1. **Identifying key terms and concepts related to the topic:** Identical to step 1 of the academic research, this step aimed to find the main terms and concepts that are relevant to the topic of web hosting performance measurement in virtualised environments. Some examples of these terms include web hosting, performance measurement, virtualisation, web development, server administration, DevOps, cloud computing and containerisation.
2. **Searching for relevant literature:** In this step, the key terms and concepts identified in step 1 are used to conduct a search for relevant literature. There were a variety of sources that were used for this purpose such as public libraries, Google books, blogs from industry experts or websites from companies offering web hosting services. When searching for professional literature it was important to use very specific search terms to identify literature that is actually aimed at people who are evaluating or running web hosting.
3. **Reading and evaluating the literature:** After identifying a set of relevant literature, the next step was to read and evaluate the literature. This included reading blog posts or website content related to web hosting performance measurement in virtualised environments. It was, of course, also important to evaluate the credibility of sources by looking at authors' publications, qualifications or company affiliations.
4. **Organizing selected literature:** After identifying the most relevant literature, the next step was to organise the literature into categories based on the main themes that emerge from research. This included categories such as web hosting architectures, high performance web development, server performance optimisation, common software configurations used in web hosting environments, cloud computing and web performance measurement.
5. **Writing the Literature Review:** The literature review summarises the best practices and important insights related to benchmarking performance in virtualised environments based on professional sources identified during research process described above. The goal was to provide comprehensive overview of current state-of-the-art approaches related to benchmarking performance in virtualised environment highlighting main findings from research conducted by professionals working with web hosting technologies every day.

3.2.2 Expert Interviews

Expert interviews are a commonly employed method of data collection in empirical social research due to their ability to serve an exploratory function [4, p. 37]. In both quantitative and qualitative research projects, expert interviews can be used to gain an initial understanding of a new or complex subject area, to clarify the research

question, or as a precursor to the development of a structured research protocol [4, p. 37]. Through the use of exploratory interviews, researchers can systematically structure their understanding of the research topic and generate hypotheses for further investigation [4, p. 37]. In this way, expert interviews can be a valuable tool for researchers seeking to gain a deeper understanding of a given subject and to develop a sound research plan [4, p. 37].

Exploratory expert interviews are conducted with industry professionals to complement the theoretical approaches developed through the literature review with practical experiences and best practices. According to [4, p. 7], engaging in conversations with experts during the initial stages of a project can help researchers avoid unnecessary time and effort spent on gathering information. These early expert interviews can provide a wealth of information in a short amount of time compared to other methods such as participant observation, field study, or systematic quantitative investigation, which can be more time-consuming and costly to organise and implement. Consulting experts in the early stages of a project can also help to avoid a considerable amount of time and effort spent on gathering information. In the early stages of a theoretically still poorly pre-structured and informationally underutilised investigation, expert interviews can provide a large amount of data compared to methods such as participant observation, field study, or systematic quantitative research, which can be more time-consuming and costly to organise and conduct [4, p. 7].

The description, introducing and process of selection of the methods and interviewees are described in greater detail in Chapter 5.

3.2.3 Benchmark Analysis

The process of benchmark code analysis on finding an approach to benchmark limited web hosting environments, using only PHP and MySQL functions, can be broken down into the following steps:

1. **Identifying potential benchmarks:** This step involves researching existing benchmarks that are relevant to the topic of limited web hosting environments. This can be done by searching for benchmarks in search engines and public source code repositories such as Google Scholar, GitHub, or other open-source code libraries and archives.
2. **Evaluating the suitability of potential benchmarks:** After identifying potential benchmarks, the next step is to evaluate them to determine their suitability for the objectives of this thesis. This includes evaluating the benchmark's methodology, the types of tests it performs, and whether it is suitable for use in a limited web hosting environment.
3. **Analyzing the benchmark:** Once a suitable benchmark has been identified, the next step is to analyse the benchmark's code. This includes understanding the

benchmark's architecture, the types of tests it performs, and the specific PHP and MySQL functions that are used. This step is important to understand how the benchmark works and how it can be adapted to suit the objectives of the thesis.

4. **Identifying limitations and areas for improvement:** During the code analysis, it is important to identify any limitations or areas for improvement in the benchmark. These can include limitations in the benchmark's methodology, the types of tests it performs, or the specific functions that are used.

In summary, the benchmark code analysis in this thesis aims at finding an approach to benchmark limited web hosting environments, involves identifying potential benchmarks, evaluating their suitability, analysing their code and identifying limitations and areas for improvement.

3.3 Challenges

Creating a synthesis of the three different research methods used in this thesis was very challenging. This is due to the fact that each method has its own advantages and limitations and provides different types of data and results. Integrating the results of these three different research methods was problematic because they were not directly comparable and therefore required different types of analysis, as explained in the sections above. In addition, the different assumptions and limitations of each method made it difficult to produce a consistent and coherent synthesis that would provide comprehensive and informative conclusions.

Chapter 4

Literature review

4.1 State of the Art

This chapter presents a theoretical examination of the architecture and design of web hosting environments, drawing upon relevant technical literature targeted towards server administrators and DevOps professionals. The focus is on current best practices, commonly employed software and configurations, as well as the technical requirements, limitations, and opportunities inherent in these environments.

The goal of this thesis is to establish a practical foundation for the technological evaluation of web hosting services. Although there has been previous academic research on this topic, it has not specifically addressed shared and managed or virtualised hosting environments. Previous studies have either focused on developing theoretical frameworks or have been limited to servers with root access or specific hardware/OS configurations. This chapter aims to provide a theoretical technological foundation for the evaluation of web hosting services in virtualised, shared and managed environments.

The literature review begins with an overview of the current state of web hosting performance benchmarking. This includes a discussion of the various types of benchmarking tools available, their features and capabilities, and how they are used to measure and compare the performance of different web hosting environments. The review also covers the various metrics used to evaluate web hosting performance, such as response time, throughput, latency, and scalability.

Next, the literature review examines the various approaches used to benchmark web hosting performance. This includes a discussion of both manual and automated methods for collecting data from web servers and analyzing it to identify areas for improvement. The review also covers techniques for simulating user traffic in order to accurately measure server performance under real-world conditions.

Finally, the literature review looks at existing research on web hosting performance benchmarking. This includes studies that have been conducted in both academic and industry settings, as well as surveys that have been conducted to gain insights

into how organizations are currently measuring and managing their web hosting environments. The review also covers research on emerging technologies that may be used in future benchmarking efforts.

While there is a significant amount of literature available on the web hosting industry and computer benchmarking in general, much of it is not directly applicable to the current research being conducted. Specifically, in the context of this master thesis on benchmarking web hosting performance in times of hardware virtualisation, the majority of the existing studies may not provide relevant insights or information for the research being undertaken. It is essential to conduct a thorough literature review to identify and evaluate the studies that are most relevant to the current research question and objectives, in order to ensure that the analysis is based on the most up-to-date and applicable information available.

4.2 The Hosting Industry

In the context of this thesis, the use of outdated data from any previous studies before 2010 must be considered not relevant for state-of-the-art benchmarking web hosting performance. This is because the technological advancements in the field of virtualisation have significantly impacted the web hosting industry, leading to extreme changes in the prices and features of web hosting packages. Therefore, using outdated data would not accurately reflect the current state of the web hosting market and its impact on performance. In order to conduct a comprehensive and accurate analysis of web hosting performance in the context of hardware virtualisation, it is necessary to gather current data through up-to-date research methods.

For example, while there is a study from 2006, which utilised electronic collection methods to create a substantial cross-sectional database that includes the prices and characteristics of web hosting packages offered in English-speaking markets [9]. However, due to the age of the article, the data is no longer current and thus, not applicable for current research. Another work confirms, that, although the significance of web applications has grown in recent years, yet the number of benchmark tools developed to evaluate the performance of the systems that host these applications remains limited [13, p. 29].

Contrary to the situation today, at the time of 2003, the vertical attributes of web hosting were fully defined and easily replicable [9]. The web hosting industry was relatively new and experiencing rapid growth during this period [9]. However, the current scenario has undergone significant changes and advancements in the field of web hosting technology, leading to a more dynamic and complex industry. This rapid evolution of web hosting industry is an important aspect that needs to be considered in any research on web hosting performance.

According to [12], in industries with differentiated products, both traditional and “new economy” firms that are vertically integrated also provide inputs to their apparent rivals in the downstream business, which creates heterogeneity between low- and high- sunk cost suppliers, and affects market entry and competitive behavior. The web hosting market is a typical example of this phenomenon, where there are primary suppliers and resellers who rent server space from them [12]. The study found that prices are sensitive to competitor clustering in quality space, which is consistent with the ease of entry for resellers with extremely low fixed costs [12].

4.3 Web Server Optimisation

Web server configuration can have a significant impact on the performance of the hosting environment because it determines how the server will respond to requests from clients. Poorly configured web servers can lead to slow response times, increased latency, and decreased throughput. Additionally, web server configuration can affect the security of the hosting environment, as certain settings can leave the server vulnerable to attack. Understanding the possibilities and limitations of web server configuration is therefore crucial to developing a working web hosting benchmark.

4.3.1 Server Configuration

There are two key configuration items for basic performance limitations on NGINX: The parameter “worker_processes” in the NGINX configuration is used to specify the number of worker processes that will be initiated [47, p. 55]. Although the default value of 1 may appear to be low, the event-driven nature of NGINX enables it to handle a substantial number of connections with a single worker process [47, p. 55]. The optimal number of worker processes is contingent upon various factors, including the number of CPU cores available on the server. As a general rule of thumb, it is recommended to set the number of worker processes equal to the number of CPU cores [47, p. 55]. Another important parameter in the NGINX configuration is “worker_connections”, which determines the maximum number of simultaneous connections that a worker process can handle [47, p. 55]. In the default configuration, this value is set to 1024 concurrent connections [47, p. 55].

The default settings for worker processes and connections settings are one of the primary challenges encountered when scaling NGINX [47, p. 445]. At the core, an NGINX worker process is responsible for handling all requests by processing them in an event-driven manner [47, p. 445]. The default settings for most NGINX installations are 512 worker connections and 1 worker process [47, p. 445]. While these defaults are sufficient for most scenarios, a highly active server may benefit from adjusting these settings to better suit the environment [47, p. 445]. However, it’s important to note that there is no universal setting that applies to all scenarios, and it’s crucial to understand the limitations and adjust the settings accordingly [47,

p. 445]. Setting the limits too high can result in increased memory and CPU overhead, which can negatively impact performance [47, p. 445].

To adjust the number of worker processes, the main NGINX configuration file must be edited [47, p. 445]. By default, the value is set to 1 [47, p. 445]. When the server is dedicated solely to running NGINX, a general rule of thumb is to set the number of worker processes to the number of CPU's available [47, p. 445]. If the server has a high number of connections and the system is not CPU bound (for example, heavy disk I/O), increasing the number of worker processes beyond the number of CPU's may improve the overall throughput of the server [47, p. 445]. Additionally, NGINX also has the capability to auto-detect the number of CPU's in the system by setting the value to auto [47, p. 445].

In summary, NGINX is an event-driven web server that is capable of handling a substantial number of connections with a single worker process. The optimal number of worker processes is contingent upon various factors, including the number of CPU cores available on the server. It is recommended to set the number of worker processes equal to the number of CPU cores, or to use the auto-detect feature. Increasing the number of worker processes beyond the number of CPU's may improve the overall throughput of the server, but it is important to understand the limitations and adjust the settings accordingly to avoid increased memory and CPU overhead. In conclusion, adjusting the number of worker processes and connections settings is essential for optimizing web server performance.

4.3.2 The PHP Workers Conundrum

A PHP worker is a background computing process that runs PHP code, which runs on the server and waits to be instructed on what to do [65]. The worker uses the available CPU cycles and RAM to execute the task and reserves some resources to stay warm for the next task [65]. Additionally, the use of dynamic PHP worker counts allows them to grow or shrink as necessary [65]. In the context of WordPress, it is mentioned that PHP workers are less efficiently handling the entirety of the request, because they are single-threaded nature of it [65]. A typical WordPress request involves passing the request to PHP, processing the code, querying the database, and passing the completed work back to the webserver, which is not optimal [65].

According to Strebel, PHP workers have recently gained a negative reputation in the WordPress hosting industry [66]. The author suggests that they are often unfairly blamed as a pretext for a host's upselling efforts [66]. The article further delves into the intricacies of PHP workers and their relevance to hosting WordPress websites by discussing an experience of unexpected interruption in the workflow caused by a prolonged loading page, which was followed by a 50x error [66]. The author expressed their concern and confusion about the incident and wonders about the possible causes and the frequency of such incidents [66].

he article emphasizes that there is no one-size-fits-all solution to such problems, and the issues could be caused by various factors throughout the hosting stack [66]. The article also highlights that the solution of “you need more PHP workers” that is often offered by WordPress hosting support desks is not sufficient and doesn’t address the underlying problem [66]. It further suggests, that the problem needs to be analyzed and addressed in a more comprehensive manner [66].

In order to understand the impact of PHP workers on site performance, Strebels colleague, Matson conducted several benchmarks [65]. The tests were conducted on a standard Amazon EC2 instance [65], and the benchmarking code used is publicly available on GitHub for replication and improvement. According to Matson, in order to accurately benchmark PHP worker behavior, the company used a simple PHP script that encrypts and decrypts a string several thousand times in order to perform CPU-heavy activities [65]. This approach is considered appropriate because it eliminates any potential noise within the tests and it allows to test PHP worker behavior by having the worker perform tasks that are heavy on CPU without impacting other factors like disk I/O or network latency [65]. In order to eliminate noise from any outbound requests and to accurately reflect different CPU core counts, Matson utilised a shell script that runs the k6 benchmarking tool against various Docker container configurations that reside on the server [65]. This approach allows running a single script that utilises different numbers of CPU cores and runs benchmarks against each separate core count [65]. The shell script used for this purpose is also available on GitHub at <https://github.com/pagely/php-worker-benchmarking>. Within the tests conducted, Matson sent a number of virtual users that match the number of cores being utilised by the testing environment [65]. For example, when running a test against an environment with 4 CPU cores, the benchmark used 4 simultaneous virtual users [65]. These users perform a request, wait for a response, then immediately send another request, and they continue doing this for a testing duration of 60 seconds [65].

The article states that a balance of PHP workers and CPU cores is what matters most [65]. Without enough dedicated CPU workers to handle the influx of traffic, requests can’t be handled efficiently [65]. In contrast, after a certain point, adding more PHP workers has a minimal impact on the number of requests per second that the server can handle [65]. Once at critical mass, there’s even a negative impact on the site’s performance as too many PHP workers are added [65]. Additionally, the data collected by the company suggests that the optimal number of PHP workers varies based on the number of CPU cores being utilised [65].

For example, in the 32 core test, the number of requests per second that could be handled steadily increased as the researcher increases the PHP workers, until it reaches 50 workers [65]. At 100 PHP workers, a slight increase was observed, then at 200 and 400 workers, a decline in performance was seen [65]. This trend was consistent across all tests he conducted (like shown in Figure 4.1).

Requests Per Second - Grouped by PHP Worker Pool Size

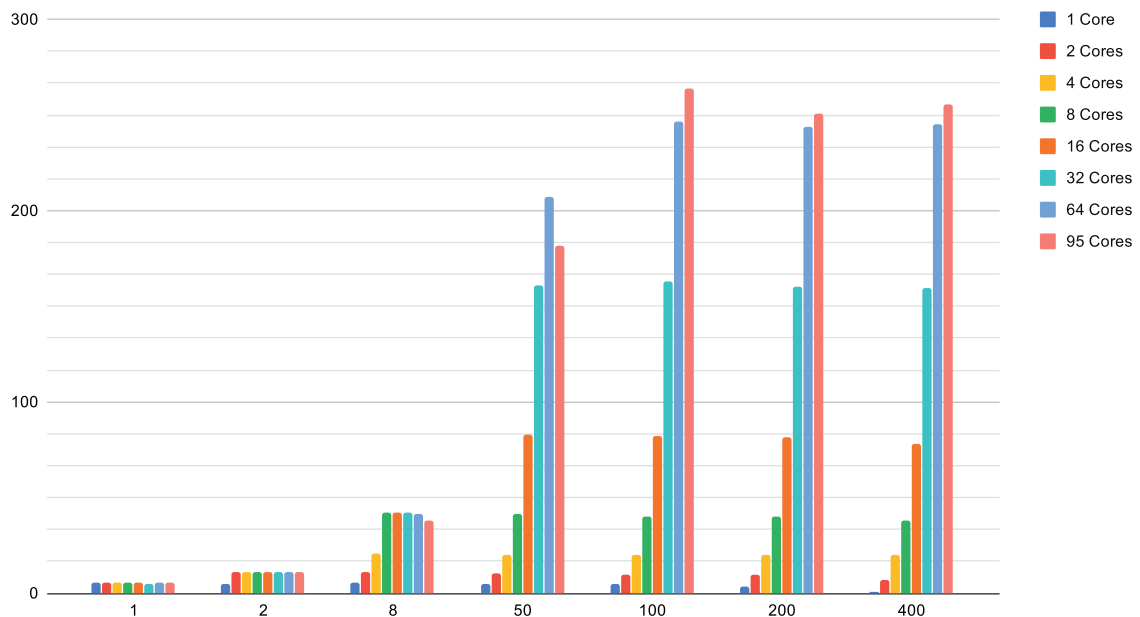


FIGURE 4.1: Requests per seconds grouped by PHP worker pool size.
Source: [65]

The article further states that after a certain point, adding more PHP workers will decrease performance overall [65]. Additionally, the impact of this phenomenon is at relatively low levels, but it can be majorly problematic for sites that don't have a dedicated resource pool [65]. Many shared/cloud WordPress hosts have thousands of people all on the same server and since the number of PHP workers will need to be significant, every site on the server is negatively impacted by the worker pool as a whole [65].

According to Matson, he also measured response times based on the number of PHP workers active for environments with different core counts in his tests [65]. The results showed that there is a measurable increase in response times as the number of PHP workers increases [65]. While the overall capacity of requests per second is a reasonable determination of how many users a site can support simultaneously, it is not the only metric at play [65]. The data showed that while the Requests Per Second statistics only change slightly between 50 PHP workers and 400 PHP workers, the request time showed an entirely different story (See Figure 4.2). The article states that when running a static count of 50 PHP workers across 32 cores, the researcher was able to make 160 requests per second, with an average response time of 309ms [65]. However, when the worker pool was increased to 100 PHP workers, the server was able to handle an additional 2 requests per second, but the average response time increased to 610ms, representing a 97% increase in the time it takes PHP to handle the request for only being able to handle an additional 1.25% more users [65].

In summary, it is stated that if one wants to increase the number of users that a website can handle simultaneously, the PHP worker count should be increased.

32 CPU Cores - http_req_waiting Times in Milliseconds

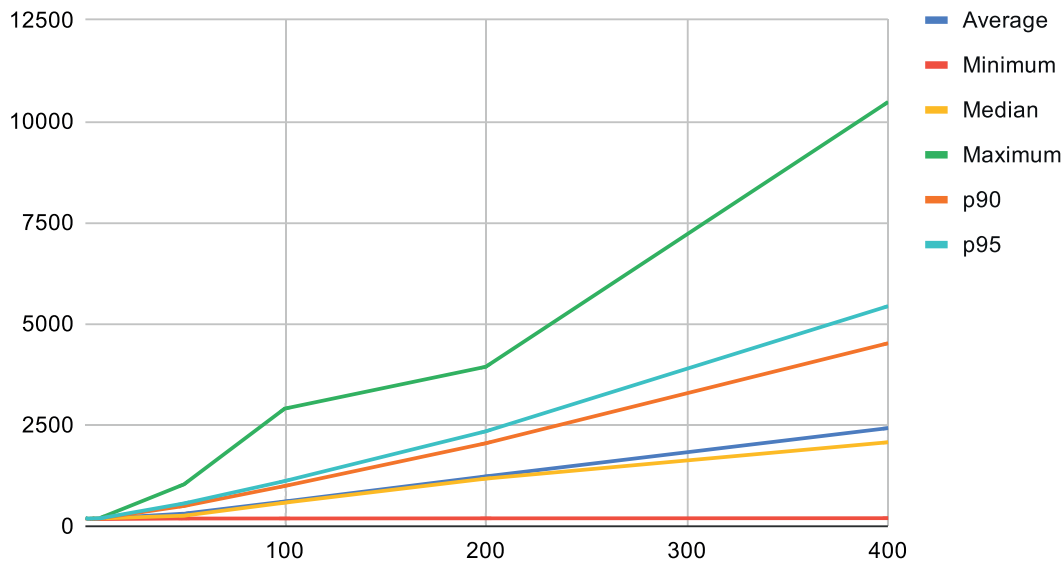


FIGURE 4.2: HTTP request waiting times in a 32-core environment.
Source: [65]

However, if the goal is to make the website faster, the worker count should be decreased. Additionally, it is recommended to use dynamic limits with a minimum and maximum worker count, as it allows to handle spikes in traffic while providing faster page load times during average traffic.

4.3.3 Performance Measurement on NGINX

NGINX is a lightweight, high-performance web server capable of serving static and dynamic content quickly and efficiently. It is also highly configurable and can be used to host multiple websites on a single server. It is also open source and free to use, making it an attractive option for many web hosting providers. It also has some unique features that should be considered when trying to measure the performance of an NGINX-powered website.

Monitoring Tools

The monitoring of performance and uptime of a web server is crucial in order to guarantee a consistent level of service [47, p. 81]. There are various methods available for monitoring these aspects, each possessing a distinct level of complexity and providing varying levels of information [47, p. 81].

But web server monitoring tools require CLI access, like the command `"nginx_status"` to enable the basic NGINX `"stub_status"` page to display rudimentary statistics and status of the service or the utility `"ngxtop"` with real-time statistics "to display useful metrics such as the number of requests per second, HTTP statuses served,

and pages/URLs served.” [47, p. 84] Therefore, these tools cannot be utilised for performance benchmarking in limited environments.

NGINX as Reverse Proxy

One of the most notable capabilities of NGINX is its functionality as a reverse proxy [47, p. 269]. Unlike a forward proxy, which is positioned between the client and the internet, a reverse proxy is situated between a server and the internet [47, p. 269]. The reverse proxy serves as an intermediary between the client and the server, and it is able to handle a wide variety of tasks such as load balancing, caching, rate limit and security enhancement by provide an providing an interface to a Web Application Firewall (WAF) [47, p. 269]. Therefore, the use of reverse proxy in NGINX can greatly enhance the performance, security, and scalability of web applications.

Content Caching with NGINX

In addition to forwarding network traffic, NGINX can also be utilised to cache proxied content [47, p. 276]. This feature enables reducing the number of requests made to the backend service, given that the requests can be cached [47, p. 276]. The caching mechanism in NGINX can greatly enhance the performance and scalability of the web application by reducing the load on the backend service, and providing a faster response time to the end-users. The caching feature of NGINX can be further optimized by adjusting the cache control headers, and implementing other caching strategies such as stale-while-revalidate and stale-if-error.

Full Page Caching

“Full page caching” refers to storing the entire webpage in a cache, and serving it for subsequent requests [40, p. 52]. This technique is more effective when the website content does not change frequently, such as on a blog with simple posts that are updated on a weekly basis [40, p. 52]. In such cases, the cache can be cleared after new posts are added. However, for websites with dynamic pages such as e-commerce websites, full page caching can create problems as the page is unique for each request and may change based on factors such as whether a user is logged in or items added to a shopping cart [40, p. 52]. In these scenarios, full page caching may not be a viable solution [40, p. 52]. Many popular platforms offer built-in support for full page caching or through plugins and modules, which take care of dynamic blocks of the page for each request [40, p. 52].

As a result the utilisation of “Full Page Caching” has a substantial effect on the response time of the PHP-Stack and should be taken into account when constructing a web hosting benchmark. The impact of this technique on observed performance is significant, and its consideration is crucial in the development of a comprehensive benchmark.

Microcaching

Caching can significantly improve the performance of a web application, however, certain situations may necessitate the invalidation of cached content, resulting in increased resource utilisation on the server, or serving stale content to the end-users [47, p. 288]. Microcaching is a technique that offers a balance between performance and functionality [47, p. 288]. By setting a low cache timeout of one second, microcaching can effectively handle a high volume of requests, such as those from a popular website [47, p. 288]. In this scenario, the majority of requests (i.e., 49 out of 50) can be served directly from cache, while only being one second old [47, p. 288]. This approach can be highly effective in reducing the load on the server, while providing a near real-time response to the end-users.

This means, that it is imperative to take into account the impact of the PHP-Stack's response time on observed performance when devising a web hosting benchmark. This factor has significant ramifications and must be given due consideration in the development process.

4.3.4 Bandwidth Management

Bandwidth management is a crucial aspect of serving large files and it's important to ensure a balance between fair distribution and performance [47, p. 372]. Proper configuration, monitoring, and adjustments are necessary to achieve this balance [47, p. 372]. Bandwidth management especially important, while serving large binary files, such as video files [47, p. 372]. This is because it is essential to ensure fair distribution of available bandwidth among users while maintaining high performance and avoiding inconvenience caused by overly restrictive settings [47, p. 372].

NGINX for example, provides several options for bandwidth management, such as rate limiting, request queue, and connection queue [47, p. 372]. These options can be configured to balance the distribution of bandwidth among users while avoiding overloading the server, resulting in improved performance and user experience [47, p. 372]. Additionally, it is also important to monitor the usage of bandwidth and adjust the settings accordingly [47, p. 372].

In the context of performance benchmarking, this means that when measuring the performance of a system or network that serves files, like a web host, it is important to take into account how bandwidth is being managed. Essentially it means that when benchmarking a system, it should be observed if the distribution of bandwidth is transparent and if it has an impact on the performance of the system.

Rate Limiting

Rate limiting is a technique that can be used to regulate the rate of incoming requests to a web application or site [47, p. 305]. It is particularly useful in situations where there is a login system or where fair usage needs to be ensured among different clients [47, p. 305]. By limiting the number of requests per IP address using

NGINX, rate limiting can help to protect the system from being overwhelmed and overloaded [47, p. 305].

This approach can lower the peak resource usage of the system and hinder the effectiveness of brute force attacks on the authentication system [47, p. 305]. Rate limiting therefore is a powerful tool that can be used to safeguard the availability and security of web applications and services. It can be implemented with various algorithms such as token bucket and fixed window, and can be configured with different parameters such as request rate and burst rate.

As an example, NGINX utilises the leaky bucket algorithm for rate limiting [49]. This algorithm is a widely accepted method in telecommunications and packet-switched computer networks for managing burstiness in situations where bandwidth is limited [49]. The leaky bucket algorithm works by maintaining a bucket with a capacity that represents the maximum number of requests that are allowed within a specified time period [49]. Incoming requests are added to the bucket, and if the bucket becomes full, the excess requests are discarded [49]. This approach effectively regulates the rate of incoming requests by controlling the rate at which requests are added to the bucket, ensuring that the system is protected from being overwhelmed by excessive traffic [49].

It is evident that this technique has a significant impact on the perceived performance of a web hosting system, particularly when subjected to load testing involving multiple simulated users concurrently requesting access to a site. This highlights the importance of considering this technique when evaluating the performance of such systems.

Connection Limiting

In addition to regulating bandwidth to ensure fair and equitable access among all users, NGINX for example also provides the capability to set limits on the number of connections [47, p. 378]. As previously discussed in this chapter, reverse proxy, rate limiting and limiting connections are distinct features, that serves different purposes [47, p. 378]. Connection limiting is particularly useful in scenarios where long-running tasks, such as downloads, are involved [47, p. 378]. Unlike bandwidth limiting, which applies per connection, connection limiting applies per IP address [47, p. 378]. However, both can be combined to ensure that each IP address cannot exceed the specified bandwidth limit [47, p. 378]. This approach can be used to improve the overall performance and scalability of web applications by effectively managing the available resources [47, p. 378].

This means, that has a built in tool, that can be used to manage the amount of data that is sent and received by different users. It can also be used to limit the number of connections that each user can make, which is useful for tasks that take a long time, like downloading. NGINX can also be used to limit the amount of data each user can

use, and both of these features can be used together to make sure that no one user is using too much of the available resources. This helps to improve the performance and scalability of web applications, but can hinder the performance of a measurement.

4.3.5 Load Balancing

Load balancing serves two primary purposes: First, to provide additional fault tolerance and to distribute the workload [47, p. 312]. This is accomplished by distributing incoming requests among one or more backend servers, resulting in the combined output of these multiple servers [47, p. 312]. By enhancing fault tolerance, the reliability and uptime of a website or application can be ensured. This is particularly relevant for high-traffic websites, where even a few seconds of downtime can cause significant disruption [47, p. 312]. Additionally, load balancing enables maintenance to be performed on the servers without affecting the availability of the website or application [47, p. 312]. In addition to fault tolerance, load balancing also allows for horizontal scaling of the website or application [47, p. 312]. This approach is more efficient than simply adding more hardware to a single server, particularly when dealing with high levels of concurrency [47, p. 312]. Using a load balancer, the number of servers can be easily increased as needed [47, p. 312]. This approach is illustrated in the figure 2.4.

In summary, because of its nature, load-balancing can be an obstacle, when it comes to measuring page speed because it distributes the load of a website across multiple servers, which can make it difficult to accurately measure the performance of a single page. Load-balancing can also cause inconsistencies in page performance, as different servers may have different levels of performance. Additionally, load-balancing can cause pages to be served from different locations, which can affect the speed of the page as well.

4.3.6 Database Caching

Query caching is a significant performance feature of MySQL that involves caching SELECT queries and their corresponding datasets in memory. This enables MySQL to retrieve the data from memory, rather than executing the query again, when an identical SELECT query is encountered [40, p. 74]. This results in faster query execution, reducing the load on the database [40, p. 74].

As a result, when building a web hosting benchmark, that includes the measurement of the database performance, it would be necessary to avoid, that performed queries can be cached.

4.3.7 CPU Sharing

While, there are several types of systems that use virtualised environments such as web hotels that host multiple websites and enterprise data centers that contain

business-critical applications [18, p. 227], these systems are usually tiered, with virtual containers hosting the applications on each physical server. A study showed, that optimizing allocated resources can reduce both operating costs and energy consumption [18, p. 227]. By hosting in a virtualised server environment, they could demonstrate that the CPU capacity allocated to the website can be optimized as the workload changes [18, p. 227]. In doing so, control structure for a processor sharing system where the allocated portion of CPU capacity can be set at fixed time intervals minimized the allocated capacity while maintaining the SLA [18, p. 227].

This means that, when using virtualised environments, like web hosting or enterprise data centers, it is possible to optimize the resources used in order to reduce costs and energy consumption. This can be done by adjusting the amount of CPU capacity allocated to each website or application as the workload changes. This can help to minimize the amount of resources used while still meeting the service level agreement.

4.4 Benchmarking in Computer Science

Benchmarking is a commonly utilised method in experimental computer science, particularly for the comparative assessment of tools and algorithms [46, p. 1]. Proper benchmarking, resource measurement and presentation of results is essential for researchers, tool developers and users as well as for tool competitions [46, p. 1]. As a result, a number of questions need to be addressed to ensure the effectiveness of benchmarking [46, p. 1]. But according to Beyer, “a number of questions need to be answered in order to ensure proper benchmarking, resource measurement, and presentation of results, all of which is essential for researchers, tool developers, and users, as well as for tool competitions.” [55] He further states, that “In most cases, benchmarks consist of a large number of tool executions, each of which is considered to be independent from the others. For accurate results, each tool execution should be performed in isolation, as on a dedicated machine without other tool executions, neither in parallel nor in sequential combination, for example to avoid letting the order of tool executions influence the results.” [55] In addition, it is uncertain to what extent the results of synthetic system benchmarks accurately reflect the actual performance of a web server: “Ultimately, [with benchmarking], you’re trying to get as close to real-world usage and workload so that you can make your architecture decisions in mind”, according to [57], which implies that there is a need for a specialised benchmark for evaluating web hosting environments.

4.4.1 The Importance of Benchmarks

Improving the performance of any server or application is affected by many different factors like the environment it is used in, its intended use, specific requirements, and the hardware components used [77, p. 155]. A common approach is to find and fix

bottlenecks in the system, which is done by testing, identifying the problem, making adjustments, and repeating until the desired level of performance is reached [77, p. 155]. In order to ensure consistency and reproducibility in testing, there is a need to automate the tests [77, p. 155]. Therefore, without an appropriate (automated) benchmarking process, it would be impossible to carry out this methodology.

4.4.2 Specialised Web Hosting Benchmarks

Although numerous benchmarks exist and “Companies such as Standard Performance Evaluation Corporation (SPEC) offer industry-standard tests” [57] they cannot be used in many cases to verify or reproduce the data for several reasons. Like mentioned in the introduction, “Benchmarking is a widely used method in experimental computer science, especially for comparative evaluation of tools and algorithms”, [55], traditional benchmarks cannot be run on shared or managed web hosting environments, as they are only allowed to run pre-installed software and modules for security and stability reasons.

Different existing operation system, drivers or applications have the different treatment methods for data; therefore it usually is difficult to measure the exact performance value. In the field of performance evaluation, there is a distinction between benchmarking tools that are optimized for desktop environments and those that are optimized for server environments. Generally speaking, most benchmarking tools are designed and intended for use in desktop environments, and while they may offer support for server workloads, they are not specifically optimized for this purpose [67].

This is an important distinction to consider when evaluating benchmarking tools, as the characteristics and workloads of desktop and server environments can differ significantly. As such, benchmarking tools that are optimized for desktop environments may not be able to accurately or effectively evaluate the performance of server environments.

Another study found out, that “most performance analysis today uses either microbenchmarks or standard macrobenchmarks ... However, the results of such benchmarks provide little information to indicate how well a particular system will handle a particular application. Such results are, at best, useless and, at worst, misleading.” [1] And the researchers “argue for an application-directed approach to benchmarking, using performance metrics that reflect the expected behavior of a particular application across a range of hardware or software platforms.” [1]

The utilisation of container technology has become a popular method for software provisioning in cloud environments due to its fast spin-up times and reduced overhead compared to virtual machine-based virtualisation [64, p. 31], that these become more and more used. The key difference between these two approaches is that while virtual machine-based virtualisation emulates the hardware and provides an

operating system as the abstraction, container-based virtualisation virtualises the operating system itself [64, p. 31]. This is achieved through the utilisation of Linux kernel primitives such as namespaces, cgroups, and layered file systems [64, p. 31]. But, as mentioned in the introduction, no real standard, objective metrics exist with which to quantify provided server resources. The concept of namespaces in Linux serves as a means of logical isolation within the kernel [64, p. 32]. This isolation is implemented at the process level, meaning that it controls the visibility of resources within the kernel for a given process [64, p. 32]. The Linux kernel acts as a guard, protecting resources such as memory, privileged CPU instructions, disks, and other resources that should only be accessible to the kernel itself [64, p. 32]. Applications running within user space can only access these resources through a system call, which traps into the kernel and executes the request on behalf of the application [64, p. 32]. As multiple user-space applications may run simultaneously on a single Linux kernel, a mechanism is needed to provide isolation between these applications. This is achieved through the use of namespaces, which create a sandboxed environment for individual applications, allowing for resources to be confined to that specific sandbox [64, p. 32]. This allows for multiple sandboxes to exist on the same Linux kernel without interfering with each other. The technique of using namespaces in Linux is a powerful tool for achieving isolation and segmentation of resources within the kernel [64, p. 32]. However, while namespaces provide isolation by restricting visibility, they do not necessarily provide true isolation [64, p. 45]. Therefore, in order to introduce resource controls for processes within a namespace, the concept of control groups, also known as cgroups [64, p. 45] is utilised. Cgroups are implemented through cgroup controllers and represented by a file system called cgroupfs in the Linux kernel, which allows for the management and allocation of resources for processes within a namespace [64, p. 45].

For example, although Beyer's framework has demonstrated its versatility, reliability and usefulness in the International Competition on Software Verification, and is widely employed by various research groups globally for ensuring dependable benchmarking [46, p. 1]. And their framework is generally able to function with a broad array of different tools, it "can (on Linux systems) currently only be done by using the cgroup and namespace features of the kernel." [46, p. 1] For these reasons, common benchmarks cannot be used for the purposes of this thesis.

4.5 Synthetic versus Real-User Performance Measurement

In general, synthetic testing is any method that uses a controlled measurement environment [29, p. 179] like in a lab. This can include a variety of different methods, such as a local build process that runs through a performance suite, load tests that are run on a staging infrastructure, or a series of geographically distributed monitoring servers that perform regular, scripted actions and record their results [29, p. 179]. These tests can evaluate various components of an infrastructure, such as

the throughput of an application server, the performance of a database, or the timing of a Domain Name System (DNS) request, and serve as a standard benchmark to identify regressions or target a particular aspect of the system [29, p. 179]. Properly configured synthetic tests can create a controlled and reproducible environment for performance evaluation, making them an ideal approach to detect and address performance degradation before it affects end users [29, p. 179].

However, synthetic tests alone are not sufficient to identify all potential performance bottlenecks [29, p. 179]. This is because the data obtained through these tests may not be representative of the many real-world factors that ultimately influence the user experience with the application [29, p. 179]. To effectively assess the performance of an application, a holistic approach is required that includes both synthetic testing and real-user measurement (RUM) [29, p. 179]. Synthetic testing, while providing a controlled and reproducible environment for identifying and addressing performance deficiencies, may not capture all factors that impact the user experience [29, p. 179]. RUM, on the other hand, captures actual performance data as experienced by users and provides a more realistic representation of application performance [29, p. 179]. Combining the data from synthetic testing and RUM provides a more comprehensive understanding of application performance and offers a more robust strategy for identifying and remediating issues [29, p. 179].

Web server benchmarking is a complex and often system- or application-dependent process that makes metrics very difficult to compare and reproduce. The ability to measure performance opens the door for improvement, but it is important to ensure that the right metrics are measured and that the process is sound [29, p. 179]. Measuring the performance of a modern web application is a complex task because there is no single metric that applies to all applications, so unique metrics must be defined for each case [29, p. 179]. Once the criteria are defined, performance data must be collected through a combination of synthetic and real-world performance measurements [29, p. 179].

4.5.1 Simulating Real World Use

The performance of web servers is usually measured in terms of: The Number of requests that can be served per seconds, the latency response time in milliseconds for each new connection or request and the throughput in bytes per second. The measurements must be performed under a varying load of clients and requests per client. Research from 2002 states, that “An important component of a Web benchmarking tool is the workload generation engine, which is responsible for reproducing the specified workload in the most accurate and efficient way.” [2] And “Distributed Web-server systems are typically subject to a large amount of traffic, which has to be reproduced somehow to evaluate their performance under realistic conditions.” [2]

This means that systems that are designed to handle a large amount of traffic on the web, such as distributed web-server systems, need to be tested under conditions that closely mimic the amount of traffic they are likely to receive in real-world use. The purpose of this is to evaluate their performance and to ensure that they can handle the expected load.

4.5.2 Understanding the PHP application stack

In order to comprehend the various metrics associated with web performance, it is imperative to first gain a thorough understanding of the underlying architecture of a PHP application stack, that is illustrated in Figure 4.3.

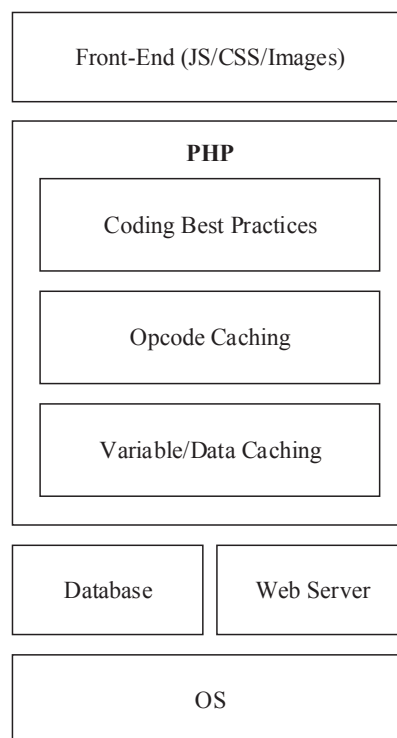


FIGURE 4.3: PHP application stack. Source: Own figure based on [24, Figure 1-1]

It is commonly acknowledged that the majority of PHP applications are presented to users via a web browser, utilizing front-end technologies such as JavaScript (JS), Cascading Style Sheets (CSS), and other front-end technologies, as well as resources such as images [24, p. 2]. The front-end interface, depicted as the topmost layer in the PHP application stack in Figure 4.3, facilitates user navigation within the web application and triggers the PHP layer [24, p. 2]. This PHP layer, in turn, comprises the business logic specific to the application and often interacts with an external storage system, such as a database or web service, to retrieve dynamic data [24, p. 2]. Furthermore, it is essential to note that all web-based PHP applications share a fundamental characteristic; they must be hosted on a web server, such as Apache or

NGINX, which is installed on an operating system [24, p. 2]. Furthermore, a database is used in most PHP-based applications, like shown in Figure 4.3.

4.5.3 Discoveries from Metrics-based Assessment

Of course, the significance of hardware in the performance of a web server cannot be overlooked [24, p. 131]. While implementing the suggestions outlined in this chapter may not yield significant improvements in performance on under-powered hardware, it is important to note that obtaining the most advanced hardware available is not necessarily the solution [24, p. 131]. Rather, it is advisable to acquire hardware within the constraints of budget and utilise the techniques to optimize its performance [24, p. 131]. Furthermore, having ample RAM enables the creation of additional processes to handle incoming requests and multiple CPUs can improve the performance of the web server [24, p. 131].

Additionally, the selection of an appropriate version of PHP for a web application can have a significant impact on the overall performance of the server. In this section, I shortly explore literature on the effects of different versions of PHP on server performance. Furthermore, I will examine the latest advancements in PHP and discuss the potential implications for server performance:

The analysis of the performance of different versions of PHP in the context of WordPress is of particular interest within the context of this thesis: Hussain (2016) performed load testing on WordPress 4, which was installed on all three different VPS [40, p. 126]. As reported in [40, p. 124], the author performed load testing on real-world applications, specifically Magento 2, Drupal 8, and WordPress 4, using three VPS configured with NGINX as the web server. One VPS had PHP 5.5-FPM, the second had PHP 5.6-FPM, and the third had PHP 7-FPM installed [40, p. 126]. The hardware specifications for all three VPS were the same, and all applications were tested with the same data and versions, thus allowing for a benchmark comparison of the performance of these applications on different versions of PHP [40, p. 124]. However, since there is no default cache embedded in WordPress and no third-party modules were installed, no cache was used. PHP OPcache was enabled during the testing. Despite this, the results were still favorable: According to the results presented in Hussain (2016), WordPress ran 135% faster in PHP 7 than in PHP 5.6, and 182% faster than in PHP 5.5 [40, p. 95].

In the meantime, however, PHP 8 is already available and PHP 8 is a major update to the PHP programming language that was officially released on November 26, 2020 [75]. It includes many optimizations and new features, one of the most notable being the Just-in-time (JIT) compiler [75]. The JIT compiler allows for the storage of native code of PHP files in an additional region of the OPcache shared memory, and the opcodes handler(s) keep pointers to the entry points of JIT-ed code [75]. According to the JIT RFC, the Just-in-time compiler implementation in PHP 8 is expected to

improve the performance of PHP [75]. However, early tests have shown that while CPU-intensive workloads may run significantly faster with JIT enabled, real-life applications like WordPress may not see a significant improvement in performance (see Figure 4.4). The RFC states that additional effort is planned to improve JIT for real-life applications, through profiling and speculative optimizations [75]. The JIT compiler allows the code to be run by the CPU directly, which can improve calculation speed, but the overall performance of web applications like WordPress also depends on other factors such as TTFB, database optimization, and HTTP requests [75].

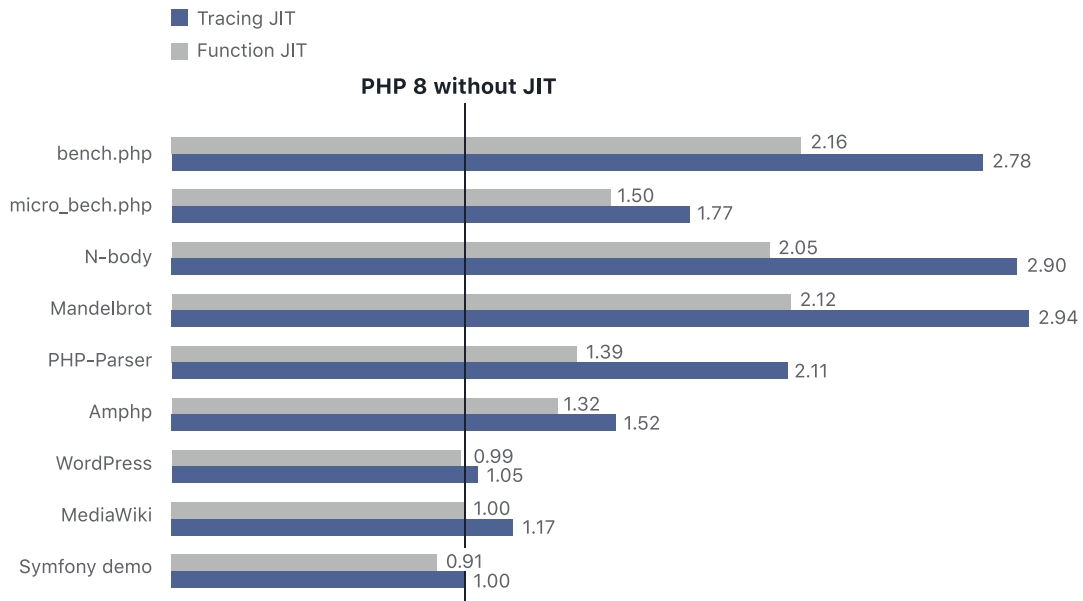


FIGURE 4.4: Relative JIT contribution to PHP 8 performance.
Source: [82]

4.5.4 Understanding Behavior of Web Servers in Stress Tests

Walkowiak conducted a study on the performance and availability of three popular web servers, Apache, IIS and NGINX, with a focus on modeling these aspects through the use of stress tests [34, p. 467]. The paper details the observations made during the tests, focusing on the behavior of the servers as they are exposed to increasing levels of request intensity. The author developed a three queue model of a web server, which could be utilised for simulating and analyzing the dependability of web systems and play a crucial role in designing an analytical model of web systems under heavy load [34, p. 467]: According the study, a simple interaction in a real system was modeled to assess the behavior of a web server. A virtual machine running an Apache server was set up as a testbed, hosting a PHP script application that could be regulated for processing time. The server was exposed to a stream of requests generated by a client application. The results of stress tests showed that the system was characterized by three distinct ranges, where response time gradually increased in the first range due to the increased probability of request overlapping (marked A in fig. 4.5), rapidly increased in the second range due to the fully utilised processing

power and overloading of web server queues (marked B), and remained constant in the third range as the server became unable to handle incoming requests (marked C). The availability plot in Figure 4.5 illustrates that the percentage of incorrect requests handled increased proportionally to the request intensity.

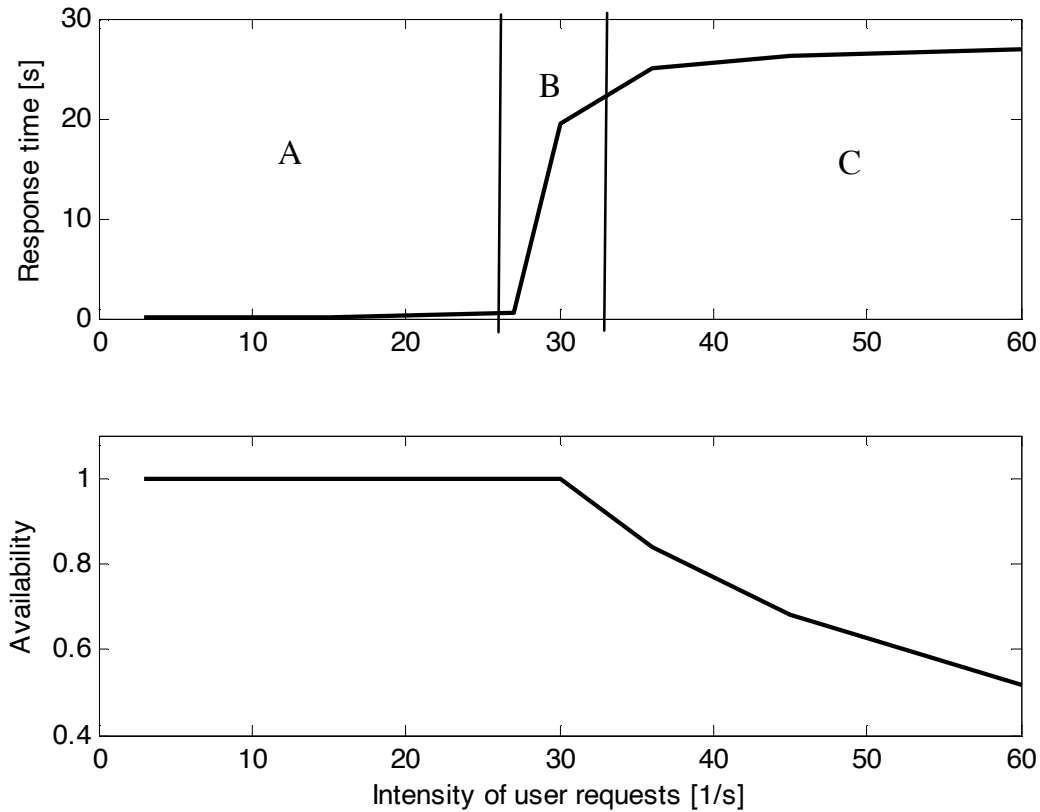


FIGURE 4.5: The performance under varying intensities of user requests. Source: [34, p. 469]

The results of the analysis led to a set of recommendations for conducting stress tests [34, p. 475]:

1. The type of client operating system should be taken into account, as it affects the limit of TCP timeouts (21 or 189 s), and therefore impacts the response time achieved.
2. Tests with a specified number of concurrent clients or a given intensity of requests should be conducted for several minutes to account for long timeouts.
3. The calculation of test results should exclude the start and end parts of results, as evidenced by the observed randomness difference.
4. The availability should be calculated for concurrent clients based on (2).
5. Tests with constant intensity of requests in range B should be repeated several times and analyzed in detail, as the server's behavior can be chaotic due to the randomness of interarrival times. This could result in the overloading of the

waiting queue, leading to the use of the retransmission buffer and causing long response times, and eventually reaching the overutilisation range (range C).

This research acknowledges the complexity of modern websites and the challenges faced by website providers in meeting the QoS requirements of clients, due to the highly dynamic and volatile nature of web traffic [34, p. 467]. The paper also notes the difficulties in analyzing, projecting, and reconfiguring web systems, given the various aspects such as hardware features, software characteristics, and multilayer architecture and network aspects that contribute to the issue [34, p. 467].

4.6 Importance of Trust in Benchmarking

The goal for a commercially successful benchmark must be to be trusted. The credibility and trustworthiness of a benchmark are essential components of its success in the market. In order for a benchmark to be considered commercially successful, it must be widely accepted and relied upon by market participants as a reliable and accurate measurement of performance. According to [69], there are a lot of trust issues within the hosting industry. And many sites use fake reviews, there was even a reported case of a web hosting provider asking its own employees to submit reviews [69]. In addition, spam networks have apparently been used in the past. In general, many marketers seem to be persuaded to favour or highlight individual companies with financial incentives [69].

Even though Ohashi understands the interests of economically active companies, his goal was to develop a non-corruptible, transparent and repeatable process. He wanted to rely as much as possible on external services that are already established in the market and therefore uses the K6 Cloud for the load test, for example [69]. According to Ohashi, the goal is to find the truth with a look at data. In other words, a quantitative approach as opposed to a qualitative one [69].

4.6.1 Limitations of Utilizing External Benchmarking

It is a common practice when setting up a website to use third-party web hosting and content delivery networks, according to [58]. Without taking this trend into consideration, any measurement study investigating the deployment and operation of websites can be greatly distorted [58]. The research community lacks generalizable tools that can be used to identify how and where a given website is hosted, which highlights the need for better understanding of this hosting trend [58].

4.6.2 CDN Usage as an Obstacle

Like explained in Chapter 2, a content delivery network (CDN) is a system of distributed servers that deliver web content to users based on their geographic location.

CDNs are used to improve the performance and availability of websites and web-based applications by reducing the distance that data must travel and by offloading traffic from the origin servers. In the context of web hosting, a CDN can be used to deliver static content such as images, JavaScript and CSS files, and media files as well as cached versions of previously generated dynamic web pages more efficiently to users. This is achieved by placing copies of these files and preprocessed web pages on servers in multiple locations around the world, so that users can access the content from a server that is physically closer to them, which can result in faster download times.

As a result, external benchmarking of a server or website that is only available through a CDN may not accurately reflect the performance experienced by its backend users, as the performance may be influenced by the CDN and other variables that are not accounted for in the benchmarking process. Therefore it is impossible to accurately measure the performance of the web server without direct access to the web server or a CDN bypass.

As a conclusion, using external load testing applications to measure pure website speed is not a feasible method for accurately assessing the performance of a host, because external benchmarking is significantly flawed due to the fact that the majority of web content is delivered by a small number of companies that provide Content Delivery Infrastructures (CDIs) such as Content Delivery Networks (CDNs) and cloud hosts. [78] The hosting industry is currently experiencing a trend towards centralisation and consolidation. A research showed, that “by focusing on the hosting industry, we show how it is heavily concentrated” “10 hosting providers account for most of the hosting for all TLDs considered.” [94] They also showed that “European ccTLDs have a strong hosting industry. However, US-based providers have been continuously conquering the market, especially in the high end of it ” [94] Currently, the majority of CDI-hosted content originates from an oligopoly of CDIs, which exert significant control over the hosting of web content. [78] The penetration of CDIs in the market has nearly doubled from 8.2% to 15% since 2015. Based on data from Alexa’s Top 1M webpages and Google CrUX’s 4.3M webpages, a CDI penetration of 24-32% was observed for popular webpages. However, 43.4% of the 392.3M page resources studied were not delivered by CDIs [78].

4.6.3 Variability in Website Speed Measurement

Earlier research indicate that the choice of service and location significantly affects the results of website speed testing [37, p. 133]. The article discusses the limitations and challenges of measuring website speed, specifically the potential inaccuracies that can occur due to inconsistent settings and the choice of service, location, and browser used for the testing [37, p. 133]. While measured values obtained by the same service and in the same browser and location slightly differ from each other, values obtained by the same service and in the same browser and location might contain a value

that deviates significantly from the rest [37, p. 133]. Values obtained by the same service, in the same browser but in different locations can significantly differ from each other, and values obtained in the same browser and location but by different services can significantly differ from each other [37, p. 133]. This poses a significant issue when comparing test results, as redesign of a website, which is a costly and time-consuming process, is often based on these results [37, p. 133]. Therefore, it is important to ensure that the results obtained are reliable [37, p. 133]. Additionally, their results suggest that there is an effect of time of day on results, which requires further research [37, p. 133].

Based on these results and confirmed hypotheses, several recommendations have been made in relation to the use of website speed tests, like, when measuring the speed of a webpage, it is important to use a testing tool that is consistent in its settings [37, p. 133]. This means that the same settings should be used each time a test is run. Additionally, it is important to run multiple tests in order to get a more accurate representation of the webpage's speed [37, p. 133]. This can be done by testing from different locations and using different browsers [37, p. 133]. When looking at the results, it is also important to exclude any values that deviate significantly from the rest and take the average or median of the remaining values as the most accurate representation of the web page's speed [37, p. 133].

Overall, this emphasizes the importance of using reliable and accurate methods when measuring website speed in order to make informed decisions about website design and optimization.

4.6.4 Cloud Infrastructure as a Black Box

It has been acknowledged, that researching servers has traditionally been a challenging task, due to the complexity of setting up representative server configurations and measuring their performance [41, p. 122]. According to [38], evaluating the performance of software systems that are deployed on black box cloud infrastructure through benchmarking can provide insight into the performance of different cloud server alternatives. However, the process can be challenging and entail a significant investment of time and resources. However, recent advancements have made it easier to benchmark servers through the availability of benchmarking software and instructions for the research community [41, p. 122]. Despite this progress, it has been noted that the existing benchmarks are often a black box, with users expected to trust the design decisions made in the construction of these benchmarks without proper justification or cited sources [41, p. 122]. This Research from 2016 shows, that in order for companies to confidently move their in-house systems to the cloud, they need to have a high level of trust in the cloud service offerings [38]. While the pricing and specifications of the cloud infrastructure are publicly available, the actual performance of the infrastructure during runtime is not known [38]. Different virtualisation technologies and multitenancy used by cloud providers can affect

the performance of software systems running on them [38]. To get a more accurate measure of cloud infrastructure performance, it is suggested that companies should benchmark their software systems running on top of the cloud infrastructure instead of relying on assumptions based on the infrastructure's price and specifications [38]. Thus benchmarking can assist companies in the initial selection phase and in the ongoing consumption phase of using cloud services [38].

4.6.5 Virtualisation and Cloudisation

Like Bose states: "The popularity of virtualised systems has engendered a need for benchmarks designed specifically to measure performance in such environments. Traditional benchmarks are typically not good choices to measure the performance of a virtual server." [14] He further notes, that "Data center consolidation, for power and space conservation, has driven the steady development and adoption of virtualisation technologies." [14] and that "This in turn has lead [sic!] to customer demands for better metrics to compare virtualisation technologies. The technology industry has responded with standardized methods and measures for benchmarking hardware and software performance with virtualisation." [14]

But as newer research showed, "performance specifications for virtual machines provided by providers are not coherent and sometimes not even sufficient to predict the actual performance of a deployment." [23] And "already on EC2, the performance indicators given by providers, namely Amazon's Elastic Compute Unit, are not sufficient to determine the actual performance of a virtual machine." [23] This means that the performance specifications provided by virtual machine providers, such as Amazon's Elastic Compute Unit, may not accurately predict the actual performance of a deployment or virtual machine. Research has shown that these specifications may not be coherent or sufficient to make accurate predictions.

Another research showed, that "Comparison of alternative cloud infrastructure offerings based only on price and server specification is difficult because cloud vendors use heterogeneous hardware resources, offer different server configurations, apply different pricing models and use different virtualisation techniques to provision them." [38] This means that comparing different cloud infrastructure offerings based solely on price and server specifications can be challenging because different vendors use different types of hardware, configurations, pricing models, and virtualisation techniques. This makes it difficult to make accurate comparisons based on these factors alone.

Therefore it is suggested that when conducting a comparison of various cloud infrastructure offerings, one should consider additional factors, in addition to price and server specifications, such as service level agreements (SLA), features, scalability, and support, in order to arrive at a more comprehensive and holistic evaluation.

4.6.6 Benchmark Development

The question of what constitutes a good benchmark has been a topic of much discussion and debate in the field. According to Huppler, who conducted a broad overview study, the information processing industry has seen the creation of numerous "industry standard" performance benchmarks in the past 25 years, some of which have been highly successful and others less so [17]. The author explored the overall requirements of a good benchmark, drawing on existing industry standards as examples. In his study, it has been acknowledged that the process of designing and implementing a performance benchmark is a complex task, requiring a balance of several competing goals such as accurate measurement, relevance, repeatability and fair comparison [17]. These goals are often in conflict with each other, making it a challenging process [17]. For example, increasing the realism of the workloads may make the benchmark more relevant but may also make it more difficult to control for external factors and produce consistent results. Therefore, designing and implementing a good performance benchmark requires careful planning, analysis and trade-offs to ensure that it effectively measures the performance of the system under test while taking into account the specific goals and constraints of the benchmarking process [17]. Like Huppler further states, "The design and implementation of a good performance benchmark is a complex process – often compromising between contrasting goals." [17]

The author also emphasizes the importance of certain characteristics that a good benchmark should possess: These include relevance, repeatability, fairness, verifiability, and economy [17]. Relevance refers to the benchmark reflecting something important and meaningful to the reader [17]. Repeatability refers to the ability to run the benchmark a second time with the same result [17]. Fairness refers to all systems or software being compared being able to participate equally [17]. Verifiability refers to the confidence that the documented result is real [17]. Economy refers to the test sponsors being able to afford to run the benchmark [17]. Huppler also notes that benchmark developers must keep these five primary criteria in mind from the beginning of the development process in order to ensure that the benchmark is effective [17].

According to Huppler, one of the key aspects of a good benchmark is its relevance [17]. This includes using software features in a realistic way, which can be challenging given the constantly evolving nature of software [17]. Additionally, a good benchmark should have longevity, with a usefulness that extends beyond just one year [17]. To be considered relevant, a benchmark should also have broad applicability and a strong target audience, who are interested in receiving the information provided by the benchmark [17]. Additionally, Huppler notes that the benchmark must be understood by the reader, use software features in a realistic way, have longevity, have broad applicability and strong target audience, and the target audience must be

interested in receiving the information [17]. The rapid pace of technological advancement within the computing industry necessitates the ongoing development of new benchmarks in order to accurately assess the performance of emerging technologies. This is a significant challenge, as stated by Huppler, who notes that the vastness and constant evolution of the computing industry requires the constant creation of new benchmarks in order to keep pace with the industry developments: “The computing industry is so vast and changes so rapidly that new benchmarks are constantly required, just to keep up” [17].

Other research states, that synthetic benchmarks are designed by combining basic computer functions in a way that developers believe will provide an indication of the performance capabilities of the machine under test [10] and that these benchmarks aim to match the average frequency of operations and operands of a large set of programs [10]. However, the author notes that the functions included in synthetic benchmarks are often artificially created to match an average execution profile, which can impair their credibility [10].

This means that the results obtained from synthetic benchmarks may not accurately reflect the performance of the machine under test, as the benchmark does not replicate the real-world scenarios the machine would be used for.

Chapter 5

Expert interviews

5.1 Methodology

As part of this thesis, expert interviews were conducted using a qualitative interviewing method with the goal of gaining comprehensive, implicit in-depth expert knowledge about the field of commercial web hosting. This approach is well-suited to the exploration of complex and multifaceted topics, and allows for the capture of rich, detailed data that can provide valuable insights into the experiences, perspectives, and expertise of the individuals being interviewed. The expert interview as a method of qualitative empirical research aims at exploring or gaining information about a specific field of action and interest [62].

By relying on expert interviews as an additional source of data, one can gain access to valuable insights and information that may not be readily available through other means, and can explore the nuances and complexities of the research topic in greater depth. However, this type of interview design also has some disadvantages: For one thing, it is very time-consuming to prepare verbatim transcripts because the two interviewees interrupt each other in conversation and relatively many sentences are not finished. In addition, the translation from German into English using automatic translation tools is error-prone due to the technical terms and colloquial language used. A structured conversation guide would have brought more structure to the conversation, but at the expense of flexibility.

Interview Structure

Based on the research goals and objectives, it appears that a semi-structured interview approach would be the most promising for this study. This method allows for the use of a predetermined protocol that includes all relevant questions, while also providing flexibility to incorporate conversational elements and probe the interviewee for additional information as needed. The semi-structured approach allows for the capture of in-depth and nuanced data while also providing structure and direction to the interview process. This method is well-suited to the exploration of complex

topics and can help to ensure that all relevant information is captured in a systematic and comprehensive manner.

According to [4, p. 7] explorative expert interviews should be conducted as openly as possible. However, for reasons of demonstrative competence, it is advisable to structure at least central dimensions of the interview process in advance in a guideline. Therefore a qualitatively oriented, guideline-structured individual interview approach is chosen for the research. This method involved the use of a structured interview protocol that included predetermined questions, but also allowed for the incorporation of qualitative data and the exploration of more open-ended topics. This approach is well-suited to the gathering of in-depth and nuanced information, and can provide valuable insights into the experiences, perspectives, and opinions of the individuals being interviewed. [4, p. 7] By using this guideline-structured approach, all relevant questions should be covered while also allowing for flexibility and the opportunity to probe for additional information as needed.

The Interviewer

The purpose of conducting an interview with a co-expert or expert of a different knowledge culture, is to establish a highly interactive and equitable dialogic exchange. By posing numerous counter-questions and demonstrating specialist competence, a symmetrical interaction was created in which both parties have the opportunity to actively contribute to the discourse and expand upon their respective areas of expertise. Through this approach, the findings of this master's thesis should be enriched and a multifaceted understanding of the topic is gained at hand. The necessary prerequisite on the part of the interviewer according to [4, p. 62] with mastery of the technical terminology is given. An institutional background can be demonstrated in the form of being a master student.

However, there is a risk of the interviewer being perceived as a publicly known critic because the author has already published a hosting comparison at <https://www.search-one.de/wordpress-hosting-vergleich/>, which could result in rejection, short answers, critical counter-questions and anticipation of questions by the expert [4, p. 62]. It should also be avoided to impose an external system of relevance on the interviewed actor, but to let him develop and formulate his own relevance [4, p. 117]. To create a sense of trust and facilitate the sharing of company internals through narrative-generating and follow-up questions, it is important for the interview style to be as open and dialog-oriented as possible [4, p. 63]. The interviewer should primarily be perceived as a recipient, with a certain information trade in terms of the potential commercial use of the generated findings in the form of a hosting benchmark being taken into consideration [4, p. 63]. This helped to create a foundation for an effective exchange of information.

Critical and Informed Perspective

The absoluteness of expert knowledge is a potential danger that should be carefully considered and addressed. [4, p. 9] To guard against the potential for blindly accepting expert knowledge as absolute, the relevant specialist literature on web hosting is considered and critically evaluated in Chapter 4. By reviewing the relevant literature and considering multiple sources of information, the pitfalls of naive belief are avoided and the interviews were approached with a more critical and informed perspective.

Expert Selection

As outlined in the previous section, expert interviews are a valuable tool seeking to gain specific insights into the thesis' subject area. The goal is to gain access to specialised knowledge and perspectives that may not be readily available through other sources. The primary focus is placed on eliciting the perspectives of experts who are actively involved in the field under investigation. Specifically, only individuals who have firsthand experience and agency within the domain under examination will be interviewed, rather than individuals who merely offer external commentary as experts. By giving voice to those who are directly involved in the field of action, this research aims to capture the most nuanced and informed understanding of the topic at hand.

According to [4, p. 116], expert status is not solely determined by the unique information an expert possesses, but also by their responsibility for making problem-solving decisions. Therefore, professional web server administrators and hosting infrastructure architects were asked about the various aspects of their work, including the construction, monitoring, and maintenance of their systems, as well as the hard and soft resource limits that are potentially in place. Additionally, it should be understood which benchmarking software is commonly used within the industry and which benchmarking software may be detected and disabled by typical web hosts.

In order to identify relevant experts, it was necessary to carefully assess their credentials and level of expertise, which can be determined through a consideration of their knowledge, skill, and practical experience in the relevant field. It was also important to note that the selection of experts was guided by the specific research question being addressed, as different experts may have different areas of expertise and may not be equally qualified to speak to all aspects of a given topic. The experts who were selected for interview in this research project were carefully chosen in order to provide a representative sample of practical insider knowledge on the topic of web hosting. These experts were seen as "crystallisation points" of knowledge, serving as representative of a larger group of actors who may have similar expertise and experience [4, p. 7].

Promoting Motivation

The process of benchmarking within a hosting environment is central to the competitive dynamics of businesses, as it pertains to the effective allocation of resources. This allocation has a direct impact on the profitability and competitiveness of the organization. Therefore expert interviews can be useful in situations where access to the field is challenging or impossible [4, p. 7].

In order to encourage high levels of participation for the interviews in this thesis, the expert interview was brought up with a potential future collaboration in the context on the commercial use of a benchmarking software as part of an online web hosting comparison website. The value of being able to objectively measure and make transparent the performance of hosting companies was emphasized, as was the possibility of using the resulting benchmark for internal performance review.

Finally, to establish trust and confidentiality, the researcher explained, that the interviewees would be anonymized in any published work and that any inadvertently revealed business secrets would be excluded from publication.

Interview Evaluation

But the careful selection of the experts from the point of view of the comparability of their positions and the presumed similarity of their experiential knowledge together with the use of the guideline instrument to ensure the thematic comparability of the experts' statements - by no means relieves the researcher of the problem of establishing and controlling the comparability of the statements[4, p. 80].

Rather than examining each text on an individual basis, this study aims to compare the expert texts with one another in order to identify overarching themes and patterns that emerge. The focus is not on the unique, individual expression of each text, but rather on the collective, shared knowledge that can be gleaned from the corpus as a whole. Through this comparative analysis, my research aims to uncover representative statements, structures of relevance, and constructions of reality that are shared across the texts. Additionally, this study aims to identify and analyze patterns of interpretation and to explore the ways in which these shape and are shaped by the collective knowledge of the experts like described in [4, p. 80].

The analysis requires the transcription of the interviews recorded as digital audio files. Since expert interviews are about shared knowledge, I consider elaborate notation systems, as they are unavoidable in narrative interviews or conversation-analytical evaluations, to be superfluous according to [4, p. 80]. Pauses, voice pitches, and other non-verbal and para-verbal elements are therefore not made the subject of interpretation. Decisions about which parts of an interview to transcribe and which to paraphrase are made in light of the guiding research questions, as described by [4, p. 80]. Thus, greetings, small talk, and parts of the conversation that deviated from

Nº	Pseud.	Position	Experience
1	Pat	Sole proprietor offering web hosting and maintenance services	> 20 years in web development and hosting
2	Max	Owner & MD of a medium-sized managed WordPress hosting company	> 5 years in web hosting
3	Andy	Customer Success Manager at a major PaaS & IaaS Cloud Hosting Provider	> 10 years in sales & IT
4	Nip	SVP Product of a major hosting company	> 10 years in web hosting
5	Tuc	Software Architect & Technical Product Owner at a major hosting company	> 25 years in web hosting

TABLE 5.1: List of interviewees.

the topic of the research question were not transcribed from the recordings, as they had no relevance to the content or contributed to gaining knowledge in the sense of this master's thesis.

The list of interviewees and the evaluation of the interviews can be found in chapter 5. Full text transcriptions of all conducted interviews are printed in the Appendix A.

5.2 Interviewees

By interviewing experts from hosting companies of varying sizes, including a major corporation, a medium-sized provider, and a one-person operation, this study aims to explore the ways in which the issue of resource control and allocation may be addressed at different levels of operation. It is possible that different approaches and solutions may be employed by these diverse types of companies in order to effectively manage their resources. Through this investigation, it is hoped that insights can be gained into the various strategies and tactics employed by different hosting companies in this regard.

In order to ensure a diverse and highly knowledgeable pool of respondents for this study, it was necessary to solicit the participation of individuals with various types of expertise from a range of companies. This approach was deemed necessary due to the specific and specialised nature of the research field, as well as the potential for a wide range of specialised knowledge to be required in order to thoroughly examine the topic at hand. By recruiting knowledge carriers from a variety of companies, I hoped that a comprehensive and well-rounded understanding of the research field can be achieved. The expert status was assumed on the specific questions within the scope of this research.

Like shown in table 5.1 the interviewees in this study are considered experts on the topic of web hosting performance benchmarking due to their varied and extensive experiences in the field. These individuals have in-depth knowledge and practical

experience in the web hosting industry, which makes them well-suited to provide insights and expertise on the topic of performance benchmarking.

5.3 Questions

Expert interviews are utilised in this study with an exploratory, field-expanding aim, in order to gather additional information including both background knowledge and practical experience. These interviews will serve to facilitate the discussion of initial approaches to benchmarking, and will allow for the identification of potential pitfalls at an early stage in the research process. By engaging with experts in this manner, it is hoped that a deeper and more comprehensive understanding of the topic at hand can be attained.

To answer the research question, answers to at least the following detailed questions must also be found:

- Which OS and software is used to run the web hosting servers?
- How are resources managed and allocated in virtualised environments?
- Which caching mechanism can be in place, that influence measured performance or observed behavior?
- How are web hosting systems monitored and what methods could be used to detect, manipulate or prevent benchmarks?
- How can soft- and hard-limits be detected and distinguished between observed short-term and long-term performance?

In order to gather relevant information for these research questions, I have developed the following questions as a guiding conversation tool and for cognitive assistance:

- Greeting and explanation of background incl. anonymisation and research objective.
- How does the physical server in a rack connect to the hosting environment for customers?
- What is the architectural approach used in the design of hosting environments, and how does it compare to the approaches used by competitors?
- What does the software stack look like, from hardware to customer account?
- Which virtualisation solution is used?
- Which operating system(s) are used?
- Which customer management software is used?
- Which web server do you use for PHP+MySQL web applications?

- How exactly are resources managed, monitored and allocated?
- How do you ensure, that one client does not compromise all others?
- What happens if a website suddenly causes a higher load than usual?
- What are the limitations in your clients hosting environments in terms of software, hardware, and their impact on the available performance?
- How can these limitations be identified and addressed?
- What strategies do you have in place to handle sudden increases in demand, such as from a television mention, with minimal warning?
- What methods do you use to differentiate between a sudden increase in legitimate traffic and a distributed denial of service (DDoS) attack??
- What server-side benchmarking features are available, and which ones are utilised or restricted by the web hosting company for their customers??
- What content management systems are most commonly used by your customers, and is the company specialised in supporting a specific CMS?
- What troubleshooting steps do you take in response to customer complaints about slow server performance or website functionality?

In the process of conducting the interviews, the preexisting knowledge from professional relationships with several companies over an extended period of time on the companies and individuals being interviewed was utilised to adapt the conversation as necessary. This may mean that not all questions were explicitly asked in some interviews. Additionally, unexpected responses has been responded to and adapted to changing circumstances in real-time.

5.4 Summarisation of the Interviews

5.4.1 Interview 1: Pat

The interview with Pat, a sole proprietor offering web hosting and maintenance services from Germany lasted 88 Minutes. The automatically translated transcript of the conversation, that was held in german, can be found in Appendix A.1.

The expert is discussing their experience with hosting virtual machines for around 14-15 years. They have been setting up a new cloud system every two years with a focus on performance and stability. They currently have two systems running, one of which is a root server that distributes the cloud system over several servers and scales itself. The newer system is a cloud system provided by a specific hosting provider, which they have been working with since the start of the provider's cloud and have optimized to ensure the necessary performance. The expert also mentions that measuring the performance of this cloud system is a complex issue because the

things displayed in the cloud do not reflect the actual server performance and can lead to unexpected slowdowns. Next, the expert discussed the use of virtualization, customer management, and web servers in the hosting environment. He describes his experience with hosting virtual machines for over a decade, and mentions using a specific cloud provider for their current systems. Pat also emphasizes the importance of remote monitoring and management (RMM) in maintaining stability and performance, and discusses using a network of servers with fibre optics to share CPU resources. He mentions that latency is not a problem in his current system, and noted that they use VMware. The expert did not mention any specific benchmarking functions they use to optimize performance. This is a standard functionality that VMware provides. It is called "vMotion" and it allows for live migration of virtual machines between physical servers without any interruption to the VM's operation. This allows for high availability and disaster recovery solutions. The use of fibre optics network between servers and the use of a management software like N-able allows for a more efficient and fast use of resources, as well as quick support in case of any issues. After that, the conversation is discussing a VMware-based system that uses a management software called N-able to manage virtual machines (VMs) in an enterprise environment. The system uses two servers that are connected via a switch with fiber optics. The servers have a file server, memory and RAM. The system is designed to run redundantly, and can migrate VMs between the servers during runtime. The VMware hypervisor, ESXi, is used to monitor processes and can recognise errors in advance and shut down the system before the error occurs. Pat mention that this type of system is not new and that many hosting providers offer similar options for shared or dedicated CPU and RAM. In the next part, the VMware product being discussed is ESXi, which is a type of hypervisor that allows for the creation and management of virtual machines. Pat mentions that the VMware system has some type of artificial intelligence that can recognise errors in advance and shut down the system before the error can cause problems. He also mentions that there are fallback and backup systems in place to prevent data loss in case of power cuts or other issues. And he mentions that they work with clients who use the system for critical processes such as time recording and documentation management, and that it is essential that the system is always available to prevent problems for the clients. We also discuss the use of different types of CPUs and how they may affect the performance of the system, with me mentioning my experience with Intel Xeon CPUs and Pat mentioning his experience with AMD Threadripper CPUs. The expert also mention the importance of being able to assign dedicated cores and a certain amount of RAM and hard disk capacity to individual VMs. Next I ask about the technical details of a virtual machine system, and the expert Pat is explaining how the system works. We discussed topics such as high-availability, assigning individual CPU cores, and the difference in performance between different CPU models. He mentions a competitor that allows customers to provide their own hardware resources for use in their cloud. Pat also mentions that they have tested the system in terms of CPU

load and bandwidth, and found that latency is the most important factor. He also mentions that they have a system for testing and rolling out updates to the main system. The further conversation is about the difficulties and considerations that come with setting up and maintaining a cloud system. Pat mentions that they have experience with setting up such systems and mentions that keeping it running is the biggest challenge. They also mention that they have a computer center in an old air-raid shelter that the customer hosts themselves and that they have their own technicians. He also mentions that they have tried to replicate this setup with other hosting providers, but have not been successful. Pat mentions that he built his first WordPress hosting from this system. After that, Pat is discussing his experience with setting up and maintaining a cloud system for hosting websites. They use VMs with Debian Linux and have automated everything using Ansible and Ansible recipes. They also use a management software called ESP Config for customer accounts and administration. They have also learned that the bottleneck for their system was not the database, but the web servers, so they have separated the web servers, database, and file system onto different servers for optimal performance. They use Nodeweb, PHP, and NGINX for the web server, and have moved away from using Apache. They also mention that most of the websites are read-only, and caching layers are used to improve performance. Pat is thinking of a different monitoring tool, such as a machine learning-based monitoring system. The system that the expert has built is a custom solution for managing and scaling resources for virtual machines. He uses Ansible scripts to automate the process of adding or removing resources, and Zabbix alerts to trigger those scripts. This allows him to respond quickly to changes in resource usage and ensure that his VMs have the resources they need. Additionally, he has a system in place to handle IP address changes when migrating VMs to different servers, using a failover IP address and tunneling traffic to the new server if necessary. This solution is used in website migrations and other cases where there may be a need to switch between different servers. The data center also has a complex network setup with multiple layers of security. The use of IP-Fire as a firewall is a legacy solution, but it has been beefed up to run redundantly and has different network levels such as red, green, and DMZ. There is also a separate internal black network that is only accessible within the data center or through a special VPN. This network is critical from a security perspective as it connects the hardware boxes and the hardware network card is completely isolated from the other networks. So essentially they have a system in place where you have two databases running in parallel, and they are constantly replicating data in real-time to ensure that if one database goes down, the other one can take over without any loss of data. After that, Pat discusses a new system that they have built for their company which includes a proxy for balancing the load on a MySQL/MariaDB database. The system includes two databases that run in parallel and replicate in real time, with a third server that only reads and backs up the database in real time. He mentions that replication only works with InnoDB for now because InnoDB writes a protocol for all commands in the database. The

interviewee also notes that there is a load-balancing system in front of the database to recognise which system is currently busy and route the call to the least busy system. Next, I raise concerns about how the system handles load balancing and replication, but Pat states that it works in any case and is extremely fast. We are discussing a database setup, specifically how to handle large data sets and prevent overloading the system. Pat explains that the goal is not to spread the load, but to minimize the risk of failure by having replication or resilience. He mentions the use of a Galera Cluster and a Galera Load Balancer, which is responsible for replication and ensuring all bookings are the same in all systems. And he also mention that many online games have the problem of loading the server or API to capacity, which can cause duplication of data. Pat also mentions that the database is set up with Zabbix and triggers to monitor network traffic and resources. I conclude, that resource management is ultimately monitoring and adding more RAM, storage, and CPUs as needed. Next Pat is discussing how they handle server configurations to ensure that a website's server does not become overloaded and cause issues for other websites on the server. He mentions that they use a script that they wrote to configure the server based on the available CPU, RAM, and other values, and that they use a monitoring system called Zabbix to monitor individual websites and their resource usage. He mentions that their main concern is bandwidth, as it is the one limitation that can cause issues for users when it becomes too full. He gives an example of a website that had a large homepage with multiple videos that caused the bandwidth to become overloaded. After that, Pat is discussing a benchmarking script that he has developed for testing the performance of hosting providers. He has contact with a specific hosting provider and has found that they have a gigabit of bandwidth internally connected with fiber optics, but it's not clear if that's the maximum capacity. The script tests four main points: CPU and memory, file system, database, and network. The speaker suggests building different scripts, including a standard WordPress installation and a PHP script that triggers typical tasks and measures how long they take. The goal is to create a benchmark that is not synthetic and focuses on the specific tasks that are relevant to the average user. He also discusses the importance of using if-else statements and generating random elements in the script. Next he discusses a self-developed CMS admin script that he built to monitor WordPress sites. He mentions that the system always reveals more information than you think and that he once displayed all the data he found in the CMS admin dashboard. He goes on to say that you can see things like how many CPUs the system has and what the load of the server is. He also discusses monitoring load on third-party systems and RAM usage on the system. He also mentions that he uses OpCache and sets limits for maximum files and the revalidate time or interval. He also mentions that they use memcached and APCu on the system. In the next part of the interview, we are discussing the use of caching systems in web hosting, specifically the use of Memcached and Redis. The expert mentions that many WordPress hosting companies are now building Redis into their stack because it reduces load in the WordPress environment. He also mentions that

APCu is the WordPress standard for saving objects and that it is anchored in the core and that it is difficult to compare systems because different systems have different configurations and limits. We discuss different tests we may conduct, such as testing PHP execution, file system reading and writing, and measuring the load on the hard disk. But maybe are not allowed to use traditional benchmarking tools because many hosting companies prevent us from using them. Next, we are discussing load testing, where a website is simulated by thousands of real users in parallel. I am aware, that this type of testing is not allowed and that some hosters might be open to it if they are informed in advance, but he also expresses that he doesn't want the hoster to know that they are being tested. Pat mentions that there may be ways to perform the tests that are not noticeable in the system and that he is willing to help with the testing, suggesting breaking it down into specific areas such as file system, database, CPU, memory, and network, and measuring the average network rate. Pat also mentions downloading the latest WordPress version and using PHP to send and receive large files to measure the throughput rate, but that it doesn't have much to do with WordPress specifically. In this section of the interview, I am discussing the strategy for building a website, specifically which hosting provider to use with WordPress or PHP. Pat advises that WordPress is not important for the construct, but I am considering using it for a practical test to measure the runtime of WordPress from start to finish. We discuss the use of caching and how it can affect the performance of the website, and Pat suggests building a plugin that calls a random URL and uses WordPress rewrite rules to bypass caching. I'm also considering using the object cache, but Pat advises that it would need to be actively activated with a plugin and potentially cleared with every call to ensure accurate performance measurements. Pat mentions that they have had issues with OPcache in the past and have implemented a plugin that clears it for updates and other triggers to avoid any issues. He suggests building different scripts to test the database, file system, and PHP, and using Ansible to automate the process. I expresses concerns about automating the process due to differences in environment variables among different hosters. We discuss potential solutions for keeping the script and WordPress version up to date, including using a WordPress remote management software or an update script that downloads a ZIP file and the importance of testing the system and the idea of taking an average of data over a period of time, possibly excluding data collected during certain times of day to stay fair. In the last part, I'm talking about my plan to build a service-based or API-based system to monitor and gather data from different hosters. And we both mention that it is important to have knowledge of the architecture and setup of the systems they will be monitoring in order to properly test and gather data. I mention that I will also be looking into existing benchmarking scripts for WordPress and other systems as part of their master's thesis, and analyzing and checking their relevance and applicability in practice. Pat, agrees with me and suggests that I don't have to reinvent existing solutions. It would be better to split up a script into smaller, self-sufficient scripts and the importance of running tests. Finally, we discuss the use

of Pats existing CMS plugin to monitor errors and performance, and discuss the use of an email function within the script. He mentions the importance of monitoring for errors and the use of a monitoring instance to terminate any processes that may be causing errors and that we need a maximum throughput per user per hour on their server to prevent someone from sending newsletters or something.

Topic	Details
Years of experience	14-15 years
Cloud Systems	Currently running 2 systems: a root server and a cloud system provided by a hosting provider (optimized for performance and stability)
Performance measurement	Complex issue due to things displayed in the cloud not reflecting actual server performance.
Virtualization	Uses VMware. The VMware system has an AI to recognize errors in advance and shut down the system before it causes problems.
Customer management	Uses N-able to manage virtual machines in an enterprise environment.
Web servers	Uses a network of servers with fiber optics to share CPU resources.
Remote Monitoring and Management	Uses RMM to maintain stability and performance.
Hypervisor	Uses ESXi (VMware) as a hypervisor to create and manage virtual machines.
CPU Models	Experienced with Intel Xeon CPUs and AMD Threadripper CPUs.
Virtual Machine System	System for assigning individual CPU cores and a certain amount of RAM and hard disk capacity to individual VMs.
Setting up and maintaining	A challenge. The expert has experience with setting up and maintaining cloud systems and has a computer center in an old air-raid shelter.
Website hosting	Uses VMs with Debian Linux, Ansible, ESP Config, Nodeweb, PHP, and NGINX. Caching layers are used for improved performance.
Resource Management	Uses Ansible scripts to automate adding or removing resources and Zabbix alerts to trigger those scripts.
IP Address Changes	Uses a failover IP address and tunneling traffic to handle IP address changes during migrations.

TABLE 5.2: Summary of the Expert Interview with Pat

5.4.2 Interview 2: Max

The interview with Max, the Owner and Managing Director of a medium-sized managed WordPress hosting company from Germany lasted 53 Minutes. The automatically translated transcript of the conversation, that was held in german, can be found in Appendix A.2.

The expert is describing their hosting environment, which includes their own hardware and software as well as third-party providers. He mentions that they have an availability of over 99.99% on an annual average, and that they are running servers in four different data centers, two of which are provided by a third-party provider and two of which are their own. The expert also mentions that they have been setting up their own hardware in data centers since 2018 and that they use VMware as their virtualization layer for the host systems, with a software-defined storage solution that is virtualised with Data Core, providing a high availability solution. In the event of a host system failure, virtual machines can be started on another host within less than a minute. The expert also mentions that the hardware is provided by a local IT systems company that has been around for 30 years and specialises in Windows servers and virtualization. After that, Max is discussing their use of VMware as a virtualization software to manage their servers. He mentions that if one host fails, another host can take over the virtual machines (VMs) from central storage, minimizing downtime. He also mentions that they have two hosts that are connected in a network and that they use Linux as the operating system on their VMs. Next, he mentions that they have 11 VMware hosts in one data center and 4 in another and that they are growing in width. Max mentions that they use Datacore, a Windows-based control software, to manage the hard disks in their servers, and that it has proven to be reliable. Then, we are discussing a high availability storage solution called Datacore that the company has been using for almost four years. Max mentions that they have found it to be incredibly stable and have decided to move almost all of their customers to it starting next year. He also mentions that they have 160 servers running in the cloud currently and plan to move those to their own data center. Then he mentions that by using their own hardware instead of renting from an external provider, they are able to offer a price cut for their servers. The company's strategy is to sell servers with less performance but at a lower price, and allow customers to upgrade individual components through the customer center. They have a "Configure Cloudserver" tariff, which is a basic configuration with 4 cores, 8 GB RAM, and 50 GB storage. Customers can upgrade the CPUs, RAM, and hard disks as needed. The cores are dynamic, with a host system that is never used more than 60% of capacity, and the company uses monitoring to ensure this and move virtual machines if necessary. The RAM and hard disk are dedicated, but the cores are not. Max explains that they used to offer dedicated cores for servers but realised it wasn't necessary for performance and was a waste of resources. They now use a strategy of monitoring the utilization of shared cores to ensure that a host system is never more than 60% busy, allowing them to invest in hardware and offer fast, expensive CPUs with with hyperthreading and that have 4GHz. They also mention that this is more efficient than other cloud providers that have cpus, that clock in the range of 2 to 2.4 GHz. Max explains that they use virtualization and virtual machines to ensure that there are always enough cores available even though they are not dedicated. Max also explains that the system is set up so that all hosts are listed through the vCenter interface and that the performance

can be viewed across all hosts, per host system, and per virtual machine. Max also mentions that the system does not actively move VMs to other hosts, but this is done manually by their team. They have a monitoring system in place that alerts them when a host reaches a limit, but this does not happen often as the distribution is evenly done. Max also mentions that there is an additional module from VMware that can do this automatically but it is costly and since their team is able to handle it manually, they have not implemented it. Max is explaining how their company uses a hybrid setup of NGINX as a reverse proxy and Apache to handle static files and PHP processing. They use PHP-FPM as a standalone module, connecting it directly to the web server. They also use a virtualization layer called CloudLinux to isolate individual WordPress websites on the same server and limit resources like CPU, RAM, and hard disk space to prevent one website from affecting others. They also limit the number of incoming requests to bypass the cache and prevent timeouts. They also have different tariff plans, one for customers with one server and one for customers with multiple servers on the same host. Next, Max is explaining that they use a hybrid mode of NGINX as a reverse proxy and Apache for static files, with the exception of PHP processing and htaccess files. They use a virtualization layer called CloudLinux to isolate each individual website from others, to ensure that one website cannot affect the performance of others. They limit the resources such as CPU, RAM, hard disks, and I/O and also limit the number of incoming requests that bypass the cache. They also limit the number of PHP threads per minute to prevent a high traffic website from taking up too many resources and causing a crash. He also mentions that when the limit is reached, the server throws a 508 Resource Limit Reached error. They also limit the number of incoming requests to bypass the cache in order to avoid server overload. They use a monitoring system that triggers an alert when the server is no longer responsive, but it does not necessarily catch when a website is slow to load. The expert also mentions that some customers, particularly agencies, may forget to install caching systems, resulting in slow loading times for their clients. In the next minutes Max is discussing the company's monitoring system for website performance and how they proactively approach customers if they notice their website is slow. They use their own monitoring system to scan all of the Plesk domains on their servers several times a day, and then proactively reach out to customers with the slowest websites to offer help with activating the company's RocketCache plugin. He also mentions that they offer a relocation service where they move the customer's website to their servers and install WP Rocket, and they also send a before and after GTMetrix comparison to the customer as part of the service. He says that they do visual tests on a few pages of the website but cannot test everything, they tell customers to check the functionality of their forms. Then, Max discusses their monitoring system for website response times and how they proactively approach customers whose websites are slow to offer solutions such as activating their RocketCache plugin. He now mentions their website relocation service where they move a website from an old host to their own, install WP Rocket

and configure it, and send the customer a before and after GTMetrix comparison. The expert notes that they prioritise functionality over performance and remind customers to test the functionality of their forms, email servers and other elements. I ask about the ability to use a PHP script to measure what's going on on their server and if customers have SSH access, the expert states that only server customers have CHRootet SSH access and that the normal customers may or may not have it, but that it wouldn't allow for much. We were discussing the limitations and challenges of testing WordPress performance. Max mentions that many WordPress plugins and performance testers are not realistic, and instead suggests measuring the response time of the backend regularly through the Admin Ajax file. He also mentions that for comparison purposes, it's important to standardize the WordPress installation on all hosts. I mention that I want to publish a plugin that allows users to compare their own system to others, but it's difficult to assume that their WordPress will be the same. Max suggests building a small PHP script to do standard tests. I ask about script runtime limits and the expert says that their standard time is 600 seconds (10 minutes) and that customers can contact support for more time. They mention that this limit is in place because many premium templates take a long time to install and they want to avoid customer frustration. I then mention that the problem with this is that if a script takes too long or runs too slowly, it can fill up the threads quickly, but the expert says that they have to test the script afterward. Finally, Max is discussing a blogger who compared different WordPress hosting providers, with one specific provider being at the top of the list. The expert then reveals that the blogger is now hosted by that provider and is receiving commission for promoting it. The expert also mentions that he found out that the blogger has set up the same WordPress installation with multiple hosting providers, but did not publish the URLs. The expert believes that measuring response time of the backend via Admin Ajax is a more accurate way to compare hosting providers, instead of using I/O benchmarks on the server. He also mentions using a service called Uptime Robot to monitor uptime and a possible issue with backups. Together we are discussing how to fairly benchmark different hosting providers by using a standard installation and measuring response times through the backend. We also discuss the possibility of running load tests to evaluate a website's ability to handle a high number of visitors, but mention the potential issue of it being considered a DDoS attack and the possibility of working with a load testing provider such as K6 to conduct the tests. We also discuss the importance of conducting a realistic test that reflects what would happen in a real-world scenario, rather than a synthetic or artificial test. Max mentions that he would be happy to provide feedback on the test procedure and that the company has a service to move customers to their service and demonstrate the performance improvements through a GTMetrix comparison and a cloned environment. The expert also mentions that this service has been successful in converting customers, with a 50% conversion rate.

Topic	Details
Hosting Environment	Own hardware and software, third-party providers
Availability	99.99% annual average
Data Centers	2 owned, 2 third-party, 4 total
Hardware	Provided by local IT company, Windows servers and virtualization
Virtualization Layer	VMware
Software-Defined Storage	Virtualized with DataCore, high availability solution
Host System Failure	Virtual machines can be started on another host in less than a minute
VMware Hosts	11 in one data center, 4 in another, growing
Datacore	Windows-based control software, managing hard disks, reliable
High Availability Storage	Datacore, used for almost 4 years, stability
Cloud Servers	160 currently, moving to own data center
Pricing Strategy	Lower price for basic configuration, upgrade individual components
Configure Cloudserver Tariff	4 cores, 8 GB RAM, 50 GB storage
Cores	Dynamic, never more than 60% utilized, monitoring to ensure
RAM and Hard Disk	Dedicated
System Monitoring	vCenter interface, performance viewable across all hosts
VM Movement	Manual by team, monitoring system in place
Hybrid Setup	NGINX as reverse proxy, Apache for static files, PHP processing
Virtualization Layer	CloudLinux to isolate individual websites
Resource Limitation	CPU, RAM, hard disk, incoming requests
Monitoring System	Alerts for server unresponsiveness
Website Performance	Proactive approach for customers if slow

TABLE 5.3: Summary of the Expert Interview with Max

5.4.3 Interview 3: Andy

The interview with Andy, a Customer Success Manager at a major PaaS & IaaS Cloud Hosting Provider from Germany lasted 55 Minutes. The automatically translated transcript of the conversation, that was held in german, can be found in Appendix A.3.

The interviewee, Andy, works for a company that focuses on software as a service, and provides infrastructure as a service and hosting. They develop their own stack and software, and their focus is on providing automated services and management for customers who want to use the advantages of a cloud-like environment on their own hardware. Andy mentions that their company is self-sufficient and all the development is done by their own team. They also operate data centers and provide hosting services, but also allow customers to use their own data centers or partners'

data centers. They have a brokerage model where customers can share their resources with others. They offer a "pay as you go" model with no packages, and their software distributes workloads automatically and can tell customers exactly how much resources they are using at any given time. They also have an API compatibility, which makes it easy for customers to switch to their services from other providers like Amazon or Azure. After that, Andy is discussing how their company's hosting service can handle sudden spikes in traffic, such as those that might occur during an episode of a popular TV show. They mention that they have experience with this and that their servers have held up well in the past. They also mention that their managed databases can automatically scale, and they are working on improving redundancy. They note that they don't have vendor lock-in, and customers can make their own backups. They also mention that they are more expensive than traditional hosting providers, and that customer is free to ask about pricing if they want to know. In summary, the company provides real cloud hosting services that are differentiated from traditional hosting companies. They offer scalable resources, pay-as-you-go pricing, and smart APIs to monitor usage. They also have a reseller model and work with system houses to provide managed services, which can save money and resources for customers. They are more expensive than some competitors but offer different services and are focused on providing modern, high-performance solutions for customers. The company is also working on improving redundancy and replication in their services. They are API-first and compatible with popular deployment tools like Terraform, making it easy to transition from other cloud providers. The company's goal is to make it easy for customers to move to a serverless, fully managed services model. As a provider of cloud hosting services, [EMPLOYER] offers a different approach to traditional hosting providers. They offer a flexible and scalable solution, with the ability to handle high traffic and unexpected spikes in usage. They are more expensive than some traditional providers, but they offer a range of services and support that make up for the added cost. They also have a new location in Amsterdam that offers more cost-effective options for customers who are price-sensitive. Overall, [EMPLOYER] is a good option for customers who want performance, good support, and the ability to scale as their needs change. Customers who are looking for a cheaper option for a small, static website may be better served by a different provider. Andy also mentions that the company's main focus is on application hosting and that they may not be the best fit for hosting static websites. The company also has a new location in Amsterdam that can offer more cost-effective options for price-sensitive customers.

5.4.4 Interview 4: Nip & Tuck

The interview with Nip & Tuck, SVP Product and Software Architect & Technical Product Owner at a major hosting company from Germany lasted 30 Minutes. The automatically translated transcript of the conversation, that was held in German, can be found in Appendix A.4.

Topic	Details
Company Focus	Software as a Service (SaaS), Infrastructure as a Service (IaaS), and hosting services with a focus on automated management and cloud-like environment on customer hardware
Development	Self-sufficient, in-house development team
Data Centers	Operate their own data centers and offer hosting services, customers can also use their own data centers or partners' data centers
Pricing Model	"Pay as you go" with no packages, smart APIs to monitor usage
Brokerage Model	Customers can share resources with others
Handling High Traffic	Experience with handling sudden spikes in traffic, servers have held up well in the past, managed databases can automatically scale, working on improving redundancy
Vendor Lock-In	No vendor lock-in, customers can make their own backups
Cost	More expensive than traditional hosting providers
API Compatibility	API compatibility with popular deployment tools like Terraform, easy to transition from other cloud providers
Goal	To make it easy for customers to move to a serverless, fully managed services model
Target Market	Customers who want performance, good support, and the ability to scale as their needs change
Main Focus	Application hosting, not the best fit for hosting static websites

TABLE 5.4: Summary of the Expert Interview with Andy

The company's web hosting platform uses in-house developments and is based on bare metal servers that are containerised using LXC containers. The web hosting stack runs on Apache with specific modules and mappings for mass management and easy domain binding. Redundancy is achieved through in-house developed block-level replication and performance is limited through soft limits set by the performance level, which determines the number of parallel PHP processes and memory usage. Static content such as images and CSS files are delivered directly and do not go through the PHP execution pipeline. The company is considering making the distinction between static and dynamic content outside of the WordPress system for optimization purposes. The experts also discussed their company's use of PHP and Linux in their hosting services, specifically in relation to WordPress and Plesk. They mention that they have built their own Linux kernel and have their own life-cycle management tool for WordPress, so they don't rely on third-party solutions. They also mention that they have an Installatron in their WordPress section but that they are not dependent on it and have alternatives, such as using an Ansible playbook or WPC-Lean. They also mention that they have used to maintain

analogues and application catalogues and packaged applications but they don't do it anymore because they found a solution that works well for them and comes at reasonably favourable conditions. They use a sharding approach, where each hosting machine is a physical machine that runs containers. This approach has the advantage of being more stable and secure, as if one machine fails, only that one machine is affected and the other machines continue to run, according to Nip. However, it also means that we can't respond as flexibly to individual customers' resource needs, as they are limited to the hardware of the current host. But this is balanced by the cost savings and stability benefits of the sharding approach. We also use tools for allocation and have security mechanisms in place to prevent abuse, such as not allowing for an immediate increase in resources for a customer. The company has tools for managing the allocation of resources and ensuring that the systems are stable. The database runs separately on a separate cluster, but it is relatively close to the network. The company also has redundancy measures in place, with each machine being available as a pair and replicated one-to-one between two data centers. The company can also handle short-term load peaks by moving individual containers to the other side, but this may come at the expense of redundancy in the short term. The company uses a combination of internal and external tools to measure performance. The internal tools include a Performance Index as a Service that monitors various values such as system utilization, health, and delivery times. They also have a small Kibana dashboard internally to compare their performance to competitors using the Google Lighthouse Score, and have test installations set up in the same way as their competitors to establish comparability. They also use external benchmarks such as Google "PageRank" to gauge their performance in relation to competitors. Tuck also mentions that it can be difficult to compare apples to oranges when it comes to performance benchmarking because competitors may use caching and optimization plugins or CDNs which can boost their indicators but are not part of the native platform performance. However, from the user's perspective, the main thing is that the website is fast. In general, the company is focused on creating transparency in performance measurements by combining synthetic and external views, and by differentiating between cached and uncached data. They have developed their own caching plugin and are working on expanding it further in future iterations, with the goal of providing a turnkey WordPress instance that doesn't require a third-party plugin. The company also wants to map the topic of CDN in their infrastructure, and are currently using Cloudflare as a CDN in their hosting environment, but are looking for a European solution in the long term. They recognise that building up their own CDN infrastructure will require more than just writing code, and will involve significant investment in infrastructure at relevant nodes.

Topic	Details
Web Hosting Platform	In-house developments, based on bare metal servers using LXC containers, Apache with specific modules for mass management and domain binding, in-house block-level replication for redundancy
Content Delivery	Static content (images, CSS) delivered directly, considering distinction between static and dynamic content for optimization purposes
Technology Stack	PHP, Linux, in-house Linux kernel, life-cycle management tool for WordPress, Ansible playbook, WPC-Lean, sharding approach (each machine runs containers)
Resource Management	Tools for allocation and security mechanisms, separate database cluster, redundancy measures (one-to-one replication between data centers), ability to handle short-term load peaks by moving containers
Performance Monitoring	Internal tools (Performance Index, Kibana dashboard), external benchmarks (Google PageRank), combining synthetic and external views, differentiating between cached and uncached data, developing own caching plugin
CDN	Currently using Cloudflare, looking for a European solution in the long term, recognition of significant infrastructure investment required.

TABLE 5.5: Summary of the Expert Interview with Nip & Tuck

5.5 Aggregated Results

The process of conducting exploratory interviews allowed for the compilation of prior knowledge acquired from professional practice, which was supplemented with the findings obtained through a comprehensive literature review, ultimately resulting in a comprehensive understanding of the subject matter.

It is commonly acknowledged that all players in the web hosting industry essentially perform the same basic functions. To optimize economic viability, it is crucial to efficiently utilize server resources while avoiding overloading and resultant failures. Additionally, strict security protocols and measures must be implemented to ensure a secure and reliable service. Depending on the specific characteristics, niche specialization, and market position, the decision-making processes can vary among web hosting companies.

A comprehensive analysis reveals that the architecture employed by web hosting companies is fundamentally similar. Typically, a host operating system, which can either be Linux-based or Windows-based, is utilized to implement virtualization. The virtualization solution employed may vary, with popular options including Hyper-V, VMWare, and Proxmox, and virtual machines are created on this host system.

The operating systems installed on these virtual machines can range from custom-compiled kernels to popular open-source distributions such as Ubuntu and Debian, to commercially available options such as CloudLinux. The surveyed companies primarily utilized the Apache web server and NGINX, with the latter utilized in some instances only as a reverse proxy. According to one provider, alternative web server LightSpeed was not deemed to be faster based on their internal testing. For customer management, billing, and user interfaces, a spectrum of options are employed, ranging from in-house development to commercial, ready-made solutions such as Plesk. Automation using tools such as Ansible is widely employed in most web hosting companies.

It is worth noting that true cloud providers, which can be considered more as infrastructure-as-a-service providers, represent an exception to this pattern. These providers can support web hosting services on their infrastructure, but the customer is responsible for adding the necessary additional layer to achieve this.

5.5.1 Limitations, Caching and Benchmarking

When measuring performance, it is important to note that all hosters limit the possible script runtime of PHP processes, and the PHP memory limit must also be taken into account. The available number of threads or PHP workers is limited on all systems. In addition, there are sometimes limitations with regard to the hard disk throughput (I/O), the number of TCP connections allowed at the same time or the database connections. Regarding the hosting of WordPress websites, the importance of a caching plugin is emphasised by all providers. Some hosting companies are developing or already offer their own, others leave the choice to the customer and some offer an included licence of WP Rocket for their customers.

All hosting companies use the internal monitoring of Linux and the web servers to monitor the load. As this can only be viewed with privileged access to the servers, it is unfortunately not suitable for the development of the benchmarking approach. All are aware that the goal should be to create a benchmark that is not synthetic and focuses on the specific tasks that are relevant to the average user. However, they also know that there are metrics such as Google's Pagespeed Insights Score or Core Web Vitals that may not reflect the true performance of a web host due to the influence of things such as themes used, plugins and website content, but are still important metrics that clients pay attention to and try to optimise for.

Common tools such as "WebPageTest.org", "GTMetrix" and "Google Pagespeed Insights" are used for external measurement of website loading times. In terms of real benchmarks, only the WP Performance Tester plugin was mentioned in the interviews conducted. When using a synthetic benchmark, the script should test four main points: CPU and memory, file system, database and network. The experts

suggest creating different test, including using a standardised WordPress installation and a PHP script that triggers typical tasks and measures how long they take.

In the literature review conducted, it was observed that most of the caching mechanisms commonly utilized by web hosting experts were mentioned in the interviews. Based on the information gathered from the interviews, it can be inferred that none of the hosting companies interviewed engage in specific monitoring with regard to benchmarks or attempt to manipulate their results, unless such actions result in improved performance of the systems.

It is important to highlight that soft limits, where requests are placed in a queue, can only be detected through extended response times, whereas hard limits can be identified through an HTTP server response status code 5xx (server error).

5.5.2 Downsides of Virtualisation

Besides all of the benefits, there are also downsides to virtualisation as results from the architectural model, which has become clear through the interviews.

- **Complexity:** Virtualisation introduces an additional layer of complexity to IT infrastructure, which can make it more difficult to manage and troubleshoot issues.
- **Performance Overhead:** Virtualisation introduces some performance overhead, as the virtualisation software must intercept and manage all communication between the virtual machines and the underlying hardware. This can lead to decreased performance, especially for resource-intensive workloads.
- **Resource contention:** Multiple virtual machines running on the same physical host may compete for resources such as CPU, memory, and storage. This can lead to poor performance and stability issues.
- **Complexity of migration:** Migrating virtual machines from one host to another can be complex and time-consuming, particularly for large and complex virtual machines.
- **Dependence on virtualisation technology:** The virtualisation technology is a key component of the infrastructure, and its failure can have a major impact on the availability of the virtual machines.
- **Lock-In Effect:** Once a virtualisation software is selected and in use, the vendor can raise the licensing costs, which can only be compensated by passing on the costs to the customers and thus leads to a competitive disadvantage.

5.5.3 Company Size and In-House Software Development

A noteworthy correlation was observed between the size of a company and the proportion of in-house software developments utilized. Specifically, it was found

that smaller companies tend to utilize commercially available off-the-shelf solutions provided by software vendors, whereas larger companies have a reduced reliance on these solutions and tend to employ their own in-house developments.

It is crucial to note that the sample size used in this study is extremely limited, and therefore, this observation should not be generalized without conducting further comprehensive research.

Chapter 6

Benchmark Analysis

6.1 Methodology

Conducting a Benchmark Analysis for finding an approach to benchmark limited (shared) web hosting environments, using only PHP and MySQL functions, is a crucial step in understanding the current state of performance measurement techniques in general and specifically for these types of environments. The process of Benchmark Analysis involves several stages, each of which contributes to the overall goal of developing a benchmark that is suitable for use in web hosting environments and that can provide accurate and reliable results.

In section 4.6.6, an examination of the fundamental concepts of benchmarks is conducted, with a focus on the characteristics that define a good benchmark.

In this section, the fundamental aspects that define a good benchmark both theoretically and practically were addressed. It was emphasized that for a benchmark to produce accurate and reliable results, it is crucial that it is not manipulable and is objective. Additionally, in order to ensure the relevance and applicability of the results obtained, it was deemed essential that the proposed benchmark be usable on all hosting packages. Further examination was conducted on the significance of practicality, specifically in terms of testing the requirements of typical websites. Through the analysis of these key elements, specific requirements were derived for a benchmark to be considered valid and useful.

The identification of benchmarks that are relevant and utilize solely PHP and MySQL functions constitutes the first step in the process. This can be achieved through a comprehensive literature review, where sources such as Google Scholar, GitHub, open-source libraries, industry blogs, review sites, and web search engines are consulted. The second stage entails a preliminary evaluation of the suitability of the identified benchmarks for the purposes of this thesis. This involves a review of the benchmark's methodology, the types of tests it performs, and its compatibility with the limited (shared) web hosting environment. In cases where a comprehensive understanding is required, the third step of the process involves an analysis of the benchmark code. This includes an examination of the benchmark's architecture, the types of tests it

carries out, and the PHP and MySQL functions that it employs. The analysis of the code is critical for comprehending the workings of the benchmark and its adaptability to the objectives of the thesis. The fourth step involves identifying limitations and areas for improvement in the benchmark, including limitations in its methodology, types of tests performed, and the PHP and MySQL functions used. This is important to ensure that the benchmark is suitable for use in the limited (shared) web hosting environment and to guarantee the accuracy and reliability of the results. Given the time-consuming and demanding nature of this task, requiring expertise in web development with PHP and MySQL, it falls outside the scope of this work.

It was established that the examination of code through a code review process is deemed necessary for the development of a tailored benchmarking suite in this study. This is owing to the requirement of a close evaluation of the functions in question to assess their impact on varying hardware and software configurations. The approach adopted for the assessment of web hosting performance encompasses the utilization of multiple measurement methods. However, a comprehensive examination of each benchmark is beyond the purview of this thesis. As such, the analysis presented here is confined to an investigation of basic methodologies and does not extend to the analysis of individual functions.

6.2 Serverside Benchmarks and Utilities

The assessment of the performance of a script or application necessitates the consideration of various elements that could impact its execution. Such elements encompass memory utilization, CPU consumption, and wall time, which refers to the complete duration for the script's completion of execution [43, p. 193]. Moreover, other factors such as disk space, network bandwidth, or API calls may also be taken into account during the performance evaluation [43, p. 193]. The specific performance demands of a script or application will dictate the significance of these elements. A decrease in any of these elements is typically regarded as an improvement in performance [43, p. 193]. It is therefore crucial to acknowledge that the measurement of performance can be a complicated process and requires a comprehensive understanding of the system and the available tools for performance evaluation.

To assist administrators in optimizing their servers, several web server monitoring and benchmarking tools are available. These tools provide in-depth information on various performance elements, such as memory utilization, CPU consumption, and wall time, as well as other factors like disk space, network bandwidth, and API calls. Administrators can utilize this information to identify bottlenecks, monitor performance over time, and make informed decisions aimed at enhancing the performance of their servers.

6.2.1 HTTP Load-Testing: Apache JMeter, Locust, Gatling

An established methodology for evaluating the performance of a web server is to use an HTTP load testing tool, such as Apache JMeter, Locust, Gatling, or any other tool that has been standardized within the organization [77, p. 155]. This is achieved by creating a configuration for the load testing tool, conducting a comprehensive examination of the web application, and running the test against the service. The review of the metrics collected from the test allows for the establishment of a baseline performance measurement [77, p. 155]. By gradually increasing the emulated user concurrency to simulate typical production usage scenarios, potential areas for improvement can be identified [77, p. 155]. It is important to note that these HTTP load testing tools are applications that require a server with root access to install programs and additional libraries. For example, Apache JMeter 5.5 requires Java 8+ [80], Locust requires Python 3.7 or later [119], and Gatling requires Java, Kotlin, and Scala [76].

6.2.2 Webserver Benchmarking Utilities: Apache Benchmark and Siege

While each layer of the PHP application stack (as illustrated in Figure 4.3) can be optimized from the front-end to the web server, it is crucial to have a tool that can measure the performance of the application both before and after the implementation of performance enhancements [24, p. 2].

Benchmarking utilities such as `ab` and `siege` belong to a class of web server benchmarking tools that provide data on the performance of a web server in response to simulated user requests [24, p. 3]. These tools enable the simulation of multiple user requests for a specific web document on a web server, including concurrent requests by multiple users [24, p. 2]. The following metrics are offered by these tools:

For instance, both tools offer information on the following metrics:

1. The duration of a single request
2. The total response size from the server
3. The number of requests a web server can handle per second

It is important to note that these tools do not evaluate the functionality of the web server, but instead measure the performance of requests for a single web document on a specific web server [24, p. 2].

The performance of a PHP application is closely linked to the efficiency of the web server software in handling incoming requests, delegating the desired actions to the PHP engine, and transmitting a response [24, p. 131]. Given the crucial role that the web server software plays in this process, it is important to minimize any unnecessary processing to prevent any degradation of the overall performance of the PHP application [24, p. 131].

6.2.3 Webserver Monitoring: Nagios, Zabbix and APM

Furthermore, there are tools that are capable of real-time monitoring of servers, such as Nagios and Zabbix (mentioned in the interview in Chapter 4), which can notify administrators of potential problems before they become critical. However, similar to the web server benchmarking utilities, these tools are only accessible to individuals with unrestricted access to the server and are not suitable for implementation in shared web hosting environments. This is due to the fact that shared web hosting environments usually restrict access to the server for security and resource management reasons.

Application Performance Monitoring (APM) tools, that can deliver server performance metrics are built to analyze the performance of a specific running application in the current system state and utilization and therefore cannot be used to objectively compare different providers, because the reproduction of this would need to simulate a specific amount of usage. Moreover, these are not available on every system/provider. [52]

While these tools should not be disregarded, a more comprehensive analysis of them is beyond the scope of this thesis.

6.3 Website Speed Measurement

It has been widely established that the speed of a website's loading time can significantly influence user engagement and satisfaction. Studies have demonstrated that even when a website provides the desired information, users tend to abandon the site if it exhibits slow loading times [5, p. 12]. This can be attributed to the annoyance caused by sluggish loading times and the presence of alternative options accessible via search engine results pages [37, p. 12].

Service name	URL
Pingdom	https://tools.pingdom.com/
WebPageTest	http://www.webpagetest.org/
GTmetrix	https://gtmetrix.com/
Dotcom-monitor	https://www.dotcom-tools.com/website-speed-test
Google PageSpeed Insights	https://pagespeed.web.dev/

TABLE 6.1: Free Online Services for Website Speed Testing, excerpt and updated URLs from "Table 2" in [37, p. 123]

It has been observed that various online services are utilized for website speed testing. Such services are characterized by features, such as measurement of page load time, analysis of website performance from various locations, and identification

of potential bottlenecks, and are widely used in the industry for the assessment of website performance. The strengths and limitations of each service, along with recommendations for the selection of the most appropriate tool for a given situation, will be discussed in Chapter 6.

Table 6.1 is an example of Website Speed Testing Tools used in academic research. Newer research from 2022 only used GTmetrix and WebPageTest along with Googles Pagespeed Insights [91, p. 116].

6.3.1 Google PageSpeed Insights

PageSpeed Insights (PSI), developed by Google, is a tool that evaluates the user experience of web pages on both mobile and desktop devices [79]. It combines lab and field data to provide suggestions for improving the page's performance. Lab data is collected in a controlled environment and is useful for identifying specific issues, but may not accurately reflect the real-world performance of the page. On the other hand, field data, which is obtained from the Chrome User Experience Report (CrUX) and captures the true user experience of the page in a live setting, has a more limited set of metrics [79].

However, despite its widespread use in the web development community and in several research studies [93, p. 177], Google PageSpeed Insights may not be appropriate for the benchmarking suite approach presented in this thesis. As per the interview with Ohasi, the metrics from Pagespeed Insights and the Core Web Vitals can only be used to a limited extent to judge the performance of a web hosting [69].

Additionally, the field data from CrUX is only available for URLs or origins that are publicly accessible, crawlable, and indexable, and have a sufficient number of samples [79]. In instances where these criteria are not met, the CrUX data for that URL or origin may not be available [79]. This data is not available for an artificial website created solely for testing purposes.

Moreover, the PSI test is conducted on a simulated mid-tier device (Moto G4) on a simulated (throttled) mobile network for mobile and an emulated desktop with a wired connection for desktop, in a Google datacenter [79]. The location of the data-center can vary based on network conditions, which can be found in the environment block of the Lighthouse report, but not selected before running the test [79]. This variability in conditions, such as network speed, server load, and location, makes it challenging for PSI to provide consistent results.

As a result, this variability in results along with missing field data makes it challenging to draw accurate and meaningful conclusions from the data obtained from PSI. Therefore, due to its lack of consistency and stability, Google PageSpeed Insights is deemed unreliable for the specific methodology employed in the thesis.

6.3.2 GTmetrix

The GTmetrix performance analysis tool, developed by Carbon60, a managed cloud provider, enables users to evaluate the speed and efficiency of their webpages [100]. The tool facilitates the performance testing process for website owners and developers by providing a user-friendly interface and comprehensive performance metrics. The GTmetrix tool assesses various performance indicators, such as load time, page size, and number of requests, enabling users to identify and address any issues that may be impacting the overall performance of their website [100]. It measures web page load time and related metrics from various locations using client browsers, allowing for comparison of request times from different geographic regions in relation to the target web server [91, p. 116]. Free users can only access testing from Vancouver, Canada using Chrome (Desktop) with an unthrottled Connection, while users who have logged in have access to 7 free test locations and paying users of the GTmetrix PRO plan have access to 15 additional “premium” test locations [101].

GTmetrix is also mentioned as a potential candidate for benchmarking purposes in expert interviews. However, the proprietary nature of its source code precludes a comprehensive evaluation of its suitability for benchmarking.

6.3.3 WebPageTest

WebPageTest is a widely used tool for evaluating the performance of web pages. The tool was initially developed by AOL and made available to the public in 2008 under an open-source license [25, p. 19]. This allows for the open-source community to contribute to the development and maintenance of the tool. The tool is accessible in multiple forms, including a public web site, an open-source project, and a downloadable version for private use. The code repository for the tool can be found on Github at the URL <https://github.com/WPO-Foundation/webpagetest>. The public website of the tool is located at <https://www.webpagetest.org/>.

The Speed Index, as introduced in WebPageTest in April 2012, is a metric that quantifies the perceived visual loading of a webpage[71]. It is calculated as the average time at which visible parts of the webpage are displayed, and is expressed in milliseconds[71]. The metric is dependent on the size of the viewport and is useful for comparing the visual loading experience of different webpages[71]. Additionally, it is recommended to use the Speed Index in conjunction with other metrics, such as load time and start render, to gain a more comprehensive understanding of a webpage’s performance[71].

In a study, which aims to conduct a performance analysis of Openlitespeed and Apache web servers in serving client requests, WebPageTest was utilized as a tool for evaluating various performance metrics [91, p. 116]. Specifically, WebPageTest is utilized for measuring bandwidth and web page load times from various files on a website server, as well as for determining the CPU usage and bandwidth utilization

on the server [91, p. 116]. The author states, that “WebPageTest is complete with a bandwidth test feature and web page load time from various files on a website server, and can also calculate how much CPU usage is on the server and the bandwidth used.” [91, p. 116]

The application operates by accepting a URL and a set of configurable parameters as input and subsequently conducting performance tests on the specified URL. Other research showed, that the flexibility of the tool is exemplified by the extensive range of configurable parameters that are available, allowing for a high degree of customization and precision in the performance evaluation process, as mention in [25, p. 19]. This robustness of configurable parameters is a notable feature of the tool and adds to its utility as a performance evaluation tool [25, p. 19].

Additionally, there are many more detailed “Page-level Metrics” available from Web-PageTest, that are metrics and measurements captured at a page-level and exposed in the JSON data[70]. The most suitable and promising metrics for evaluating the performance of a web host are:

- **TTFB (Time to First Byte):** This metric measures the time from when the navigation started until the first byte of page content is returned (after redirects).
- **loadTime (also fullyLoaded):** This metric measures the time to fully load the page in milliseconds.
- **bytesIn:** Number of bytes received (To check for compression and size optimisation)
- **chromeUserTiming.LargestContentfulPaint:** Chrome Web Vitals Time to the largest contentful paint
- **chromeUserTiming.firstContentfulPaint:** Chrome-reported first contentful paint

These metrics provide insight into how quickly the page is loading, how much data is being transferred, and the overall user experience of the website. Therefore these metrics are considered to be the most relevant for evaluating the performance of a web host in this thesis.

In summary, WebPageTest presents itself as a viable option for assessing the loading time of a standardized WordPress website in order to evaluate the relative performance of cached frontend speed in conjunction with network performance.

6.4 PHP-based Benchmarks

Additionally, there exist various tools and libraries, written in PHP, that facilitate the assessment of the performance of a shared hosting system.

6.4.1 Torrisi's PHP Benchmark Performance Script

In 2010, Alessandro Torrisi built his "PHP Benchmark Performance Script", which was released under Creative Commons CC-BY license and therefore could also be used in a commercial project in the future [22]. His PHP script (See Appendix B.1) is a basic benchmark test that measures the performance of various functions and operations in PHP. The script defines four functions: "test_Math", "test_StringManipulation", "test_Loops" and "test_IfElse", which test the performance of mathematical functions, string manipulation functions, loops and if-else statements respectively. Each function takes an optional parameter "count" which specifies the number of times the operation is to be performed. The script starts by measuring the current time using the "microtime()" function, and then performs the operation the specified number of times. After the operation is complete, the script calculates the time taken by subtracting the start time from the current time and returns the result in seconds with three decimal places using the "number_format()" function. The script then calls all the functions, that start with "test_" and prints out the time taken for each function to execute, as well as the total time taken for all the tests. It also prints out some information about the server, PHP version, and platform (See listing in B.1).

However, it is crucial to note that the aforementioned script represents only a rudimentary benchmark test, which primarily evaluates the performance of various functions and operations in PHP. It is important to acknowledge that such basic tests only furnish limited understanding of the actual performance of a web host when executing more complicated functions. For example, the primary memory utilization is minimally impacted during the processing of the functions, and there are no write or read accesses to the file system or database, which are pivotal for the execution of a content management system such as WordPress.

6.4.2 TiM International's Web Hosting Performance Benchmark

"Web Hosting Performance" is an independent research project that aims to provide benchmark testing and performance monitoring of web hosting providers [104]. The goal of this project, according to the owners, is to review and evaluate the performance of various web hosting providers in order to promote competition and address both good and poor performance [104]. The project was originally initiated as a hobby project by T. Almroth, the founder and developer of LiteCart [104]. They further state, that there have been instances where web hosting providers have marketed themselves as offering exceptional speed, but upon further testing, it was discovered that their performance was only a fraction of that of their competitors, despite charging significantly higher prices [104].

The project offers different free so-called "satellites" testing tools, that monitors the performance of a website and reports it back to the project [106]. By its approach, the benchmark data generated by this project should reflect the relative performance

<i>Operation</i>	<i>Measurement Description</i>
CPU	
Pi	Time elapsed for calculating pi in milliseconds.
Disk	
Read	Amount of data read per second in MegaBytes.
Write	Amount of data written per second in MegaBytes.
Compile	Amount of data compiled per second in MegaBytes.
MySQL	
Connect	Time elapsed for connecting to MySQL server in milliseconds.
Insert	Number of inserted rows per second.
Select	Number of read rows per second.
Search	Number of read rows per second.
Update	Number of updated rows per second.
Delete	Number of deleted rows per second.
Fsockopen	
Upstream	Amount of data sent per second in Megabits.
Downstream	Amount of data received per second in Megabits.

TABLE 6.2: Web Hosting Performance Metrics [105]

of the web servers, rather than variations in geographical location or local network conditions [104]. The benchmarks requires PHP 5.3+ along with MySQL 5.5+ and the "BCMath Extension", a default PHP extension on common systems, like Debian Linux [103] and CloudLinux [102].

This benchmark presents a viable option for assessing the fundamental performance capabilities of a server utilizing granular Web Hosting Performance Metrics within various relevant domains, as depicted in Table 6.2.

6.5 WordPress-based Benchmarks

This section aims to provide an overview of the various WordPress plugins that are specifically designed to measure web hosting performance. These plugins, such as "WPPerformanceTester" by Kevin Ohashi [87] and "WordPress Hosting Benchmark tool" by Anton Aleksandrov [73], offer an efficient and convenient method for assessing the capabilities of web hosting servers in a WordPress context. These tools employ a variety of performance metrics, including measures of CPU and memory performance, filesystem performance, database performance and network speed, to

provide a comprehensive evaluation of the server's capabilities. This section will discuss the features and methodology of these WordPress-based benchmarks and provide a comparison of their performance assessments.

6.5.1 WordPress Hosting Benchmark tool by Anton Aleksandrov

This WordPress benchmark aims to investigate the impact of raw server power on the speed of a WordPress website [95]. According to the author, this plugin has been developed to conduct unified tests and measure the execution times of various performance metrics [95]. Based on these results, a score will be assigned to the server performance, enabling a determination of whether the server's capabilities are sufficient for a given project. The plugin should enable the comparison of different servers or hosting platforms by measuring the time taken for each test [95]. The tests are grouped and target various components of the server including CPU, memory, filesystem, database performance, WordPress object cache, and network speed [95].

The testing methodology employed includes assessments of CPU and memory performance, as well as filesystem and database performance [95]. The CPU performance is evaluated through the utilization of checksum calculation via the md5 function, in conjunction with random number generation and regular expression functions, which exert a heavy load on the processor [95]. Memory performance is assessed by creating a large in-memory array filled with random data and performing trivial operations on its elements [95]. Additionally, as of version 1.2 (released on April 27, 2022) the plugin includes reverse connectivity tests which measure the connection time when connecting to a WordPress site [95].

Filesystem performance is evaluated through writeability testing, in which attempts are made to write a 50MB file 20 times, as well as small file IO testing, which involves writing 150Kb files 500 times [95]. Additionally, a file copy benchmark is conducted, in which a 50MB file is copied locally 20 times [95].

Database performance is evaluated through a series of tests, including the insertion of a large amount of random data into temporary tables, the execution of simple select and update queries, and the execution of more complex select queries utilizing table joining operators.

Network speed is evaluated through a test that attempts to download a large file from a global CDN, which serves as a proxy for measuring the speed of the server's internet connection [95]. It is important to note that this test measures the tendency and quality of the internet connection, however it is not a direct measurement of the server's network speed [95].

It should be noted that this plugin generates large temporary files and runs a significant number of SQL queries in order to assess the performance of the web hosting server [73]. It is important to ensure that the hosting has a minimum of 500MB of free disk space and that the database server is configured to accommodate a large

number of queries [73]. All temporary files and database tables generated during the benchmarking process will be deleted upon completion of the assessment [73].

These tests are conducted during local benchmarks and for a short time after, allowing for the analysis of the impact of a single heavy task on page load time [73]. It should be noted that this plugin does not require any additional software or tools as all tests are executed from within PHP [73].

6.5.2 WPerformanceTester by Kevin Ohashi

“WPerformanceTester” is open source software that was developed to conduct benchmarking of WordPress in the context of the WordPress Hosting Performance Benchmarks study conducted by Review Signal in 2015 [87]. The tool is designed to evaluate the performance of a server by “stressing its PHP, MySQL and running \$wpdb queries” [87].

WPerformanceTester conducts a series of tests, including: [87]

- **Math:** 100,000 math function tests
- **String Manipulation:** 100,000 string manipulation tests
- **Loops:** 1,000,000 loop iterations
- **Conditionals:** 1,000,000 conditional logic checks
- **MySQL** (connect, select, version, aes_encrypt): Basic MySQL functions and 5,000,000 AES_ENCRYPT() iterations
- **\$wpdb:** 250 insert, select, update and delete operations through \$wpdb

In addition to the tests, the tool also allows for comparison of the server’s performance against an industry benchmark, which is calculated as the average of all submitted test results [87]. This feature provides insights into how the server’s performance compares to the industry average [87].

According to Ohashi, it is important to note that performance can be measured in a variety of ways and that “WPerformanceTester” is just one component of a larger performance benchmark [87]. It tests a single server (or node) that it is running on, and may not provide insight into how well a clustered or distributed setup performs [87]. He further states, that the tool focuses on the raw speed at which a system can execute code and perform database operations, and that it is important to note that real website performance is not always correlated with raw speed [87]. For an example, Ohashi mentions, that caching layers can greatly impact website performance, and even a slower website may have a faster result [87]. Ohashi therefore suggests that his “WPerformanceTester” should be used as one tool among many in measuring performance and that other tools should be used to test other facets of performance. Finally, Ohashi warns that the script may time out (max_execution_time limit) if it

does not show any results, and suggests increasing the `max_execution_time` in the `php.ini` to solve this issue [87]. But Ohashi also states, that some plugins may also cause “WPPerformanceTester” to run exceptionally slow, and thus make it more likely to hit this limit [87]. His solution to that issue is to temporarily disable plugins that might be interfering with the plugin [87]. **While it may be handy to easily compare a web hosting package to industry benchmarks with that plugin, this issue arise the question, if WordPress is the proper environment for running synthetic benchmarks.**

The main PHP Script to benchmark PHP and MySQL-Server is inspired by the “php-benchmark-script” by Alessandro Torrisi, which I analyzed in 6.4.1. The techniques and principles used in this benchmark are identical [86] but Ohashi added database performance tests for MySQL via `mysqli` and `$wpdb`.

6.5.3 WordPress as an Environment for Synthetic Benchmarks

An investigation was carried out to examine the suitability of the WordPress environment for running synthetic benchmarks and the impact of variations in themes, plugins, and content on the results of benchmarking plugins within this environment. To do so, a simple yet effective experiment was devised.

Two separate WordPress instances were installed on the same server, both utilizing the same shared hosting package, with the intention of theoretically producing identical results for both benchmarks. However, the two websites in the two instances were configured differently, using different themes, plugins, and possessing distinct content in their databases and file systems. This experimental design enabled an assessment of the impact of these variations on the benchmarking results.

The results of the investigation revealed that there was no significant impact on the results of the benchmarking plugins due to the variations in themes, plugins, and content.

The results of the performance tests conducted on the domains “search-one.de” and “webmasterpro.de” using the “Wordpress hosting Benchmarking Tool” revealed server scores of 6.4 and 6.5 for the former, and 6.5 and 6.4 for the latter, respectively (see Figure 6.1). Despite slight variations in the granular test scores, it was observed that these fluctuations were consistent within each individual test and domain, leading to the conclusion that the performance is not impacted by factors such as themes or plugins. It is hypothesized that this consistency in performance is likely due to the fact that the “Wordpress hosting Benchmarking Tool” plugin executes the tests from the backend via AJAX (using the JavaScript library jQuery), thereby avoiding the potential for variations in elements such as themes or plugins to affect the results. This assumption was further validated through an analysis of the plugin’s architectural code [74] (see listing in 6.5.3).

```
<?php
```

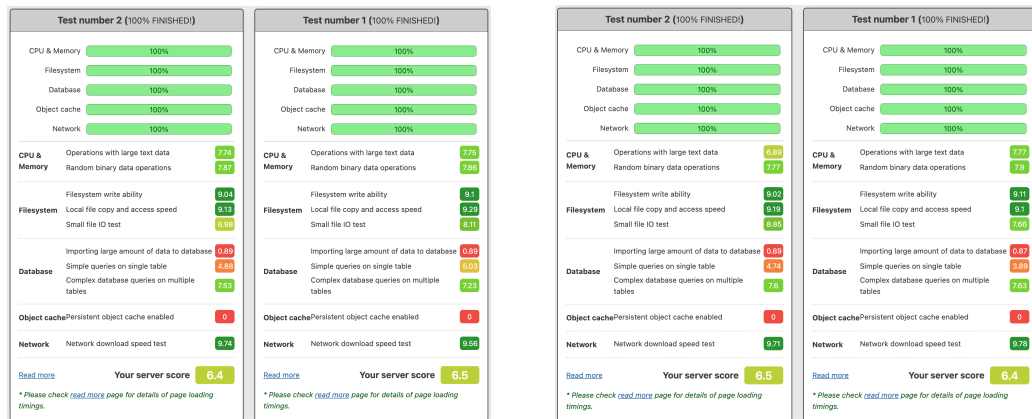


FIGURE 6.1: Test results from Wordpress hosting Benchmarking Tool.
Source: Own Figure (Screenshot)

```

/**
 * @package wp benchmark io
 */
/*
Plugin Name: WordPress Hosting Benchmark tool
Plugin URI: https://wordpress.org/plugins/wpbenchmark/
Description: Utility to benchmark and stresstest your Wordpress hosting server, its capabilities, speed
and compare with other hosts.
Text Domain:
Version: 1.3.2
Requires PHP: 5.6
Network: true
Author: Anton Aleksandrov
Author URI: https://wpbenchmark.io
License: GPLv3
License URI: http://www.gnu.org/licenses/gpl-3.0.html
*/
(...)
jQuery.ajax({
    url:'',
    type:'POST',
    data: {
        start_benchmark:1,
        _wpnonce:'".wp_create_nonce("wp-benchmark-io-start-new-bench")."',
    },
    dataType:'json',
    success:function(data) {

        if (data.status>0) {
            jQuery('#wpio-start-btn').html('Running...');
            set_bench_code(data.bench_code);

        } else
            jQuery('#wpio-start-btn').attr('disabled', false).html('Error occured, try again'
            );

        //if (data.status<0)
        show_output(data.description, data.status);

        create_progress_info(data.group_progress, data.skip_object_cache_tests);

        if (data.progress<100 && data.status>0)
            setTimeout(run_benchmark, 1000);

    },
    error:function(data) {
        show_output('Unknown error occured during connecting to plugin backend!', -1);
        console.log(data);
        jQuery('#wpio-start-btn').html('Ready to run again!');
        jQuery('#wpio-start-btn').attr('disabled', false);
        jQuery('#wpio-save-setting-btn').attr('disabled', false);
    }
});

```

```

    }
  });
}
(...)
?>

```

6.6 Load-Testing and Stress-Testing Services

A similar approach to the server-side load testing tools described in 6.2.1 is taken by the commercial online load testing services (like k6 by Grafana Labs). Load Testing is a type of Performance Testing used to determine a system's behavior under normal and peak conditions by simulating concurrent users or requests per second[107]. It is used to ensure that the application performs well when many users access it at the same time. It is used to assess the current performance of a system and ensure that it continues to meet performance standards as changes are made[107]. Performance goals are determined by understanding the average and peak traffic on a system and configuring the load test with the peak load in mind[107]. Of course, these data can also assist in evaluating performance. In his hosting comparison, Ohasi for example defined the following threshold values as minimum requirements: 99.99% uptime and an error rate of less than 1% in the load test as well as a response time of no more than 1 second [69]. But his methodology shows, that his load testing process involves a 30-minute scaling user test, with a maximum number of clients, based on pricing tiers [68]. The test simulates user actions including hitting the homepage, login page, login, and all pages and posts [68]. The test starts with 1 user and scales to 'n' users over 20 minutes with a sustained peak load for 10 minutes, but does not go to the limit of the server [68]. In order to observe the phases described in 4.5.4, and thus measure the actual limit of "still stably processable" visits, the web server must be brought to this limit and even beyond until the failure of the request.

It is important to note that in the event of system failure during a load test, the test has essentially transitioned into a stress test [107]. Stress testing is a specialized form of load testing that focuses on determining the limits of a system and its ability to handle extreme conditions [108]. It is used to verify the stability and reliability of a system under heavy load, and can be accomplished by using a tool to push the system beyond its normal operations to its breaking point[108]. The goal of stress testing is to understand how a system behaves under extreme conditions, determine its maximum capacity, identify its failure mode, and determine if it can recover without manual intervention [108]. Stress testing is often used to prepare for high-traffic events such as Black Friday or Cyber Monday, and is typically performed in a series of gradual steps, each increasing the concurrent load on the system [108]. The ultimate goal of stress testing is to identify the performance limits of a system and understand how it responds when pushed to its limits [108].

When conducting a stress test to determine the maximum capacity of a specific hosting service, it may present challenges due to the nature of the test, which simulates a high

volume of requests to the system. From the provider's perspective, this may appear as a denial of service (DoS) or distributed denial of service (DDoS) attack, and as a result, the testing requests may be blocked. This can make it difficult to accurately perform a stress test and determine the true capabilities of the hosting service.

Therefore, it is important to note that in order to avoid any misinterpretation of the testing activity as an attack, proper permissions and approvals should be obtained before conducting a stress test, and clear communication should be established with the hosting provider to avoid any potential misunderstandings or disruptions. Another important point to note is that this communication can prevent an anonymised and tamper-free test, as one identifies oneself as a tester.

6.7 Abandoned Benchmarks

In the chapter "Performance Baseline" in the Book [39, p. 85], there is a benchmark mentioned called "Serverbear". According to these information, Serverbear benchmarks tests the CPU (UnixBench score), the performance of the I/O subsystem, and the download speed from various locations to your server. ServerBear was a website and tool for evaluating the performance of web servers. However, it appears that the website is no longer accessible and the project seems to have been abandoned. This means that the tool is no longer available for use, and any information or data that may have been generated by the tool is also not accessible.

In the field of software engineering, it is not uncommon for projects to be abandoned due to a variety of reasons such as lack of funding or loss of interest from the developers. The unavailability of the tool and the fact that the project appears to be abandoned, it can be inferred that the tool is no longer maintained or supported by the developers. This limits the utility of the tool and the ability to access any data that may have been generated by the tool.

It is essential to note that, The dependability of the tools and their continuance support is crucial for any research or evaluation that may have been conducted using the tool. Therefore, when evaluating tools for a given purpose, it is important to consider the current status and long-term availability of the tool.

6.8 Client-Side Benchmarks

According to Barker, the World Wide Web Consortium (W3C) established a new working group, referred to as the Web Performance Working Group in late 2010 [25, p. 83]. The stated mission of this working group was to develop methods for measuring various aspects of the performance of user agent features and application programming interfaces (APIs) [25, p. 83]. The scope of work of the Web Performance Working Group, which was active until June 2010, included the examination and enhancement of various aspects of application performance related to user agent features and

APIs [63]. This included the measurement of network and rendering performance, responsiveness and interactivity, memory and CPU utilization, and application failures [63]. Additionally, the group aimed to deliver similar APIs and infrastructure to enable the measurement and improvement of user experience [63]. These deliverables were intended to be applicable to both desktop and mobile browsers, as well as other non-browser environments where appropriate [63]. In practical terms, this means that the working group has created an API that browsers can make available to JavaScript, providing access to key performance metrics [25, p. 83]. This API is implemented through the introduction of a new performance object, which is a native part of the window object [25, p. 83]. Specifically, the performance object can be accessed via the following JavaScript code:

```
window.performance
```

This is a JavaScript API that allows for the measurement of various performance metrics within a web browser. These metrics include things like load time, rendering time, and user interaction timing. However, this API is specific to the client-side performance of a website and does not provide information about the server-side performance of the website. Web hosting server performance includes metrics like server response time, CPU utilization, and network bandwidth usage. These metrics are not directly measurable through the window.performance API, as they are specific to the web hosting server and not the client-side browser. To measure server-side performance, one would need to use different tools such as server-side monitoring software, server log analysis tools, or synthetic monitoring tools that simulate requests to the server.

6.9 Network Performance Measurement

Furthermore, an additional method to evaluate network performance is through the measurement of upload and download speeds by transferring files of varying sizes. This method allows for the determination of the overall network performance in terms of data transfer speeds. However, it is important to note that this method is highly dependent on the file serving or receiving side, as it requires the involvement of another server. This is because the network performance of a given server cannot be accurately measured without taking into account the performance of other servers that are involved in the data transfer process.

In summary, measuring network performance by transferring files of varying sizes is a viable method, but it is critical to keep in mind that it is dependent on the performance of the server that is serving or receiving the files. As such, it is important to consider the performance of the entire network when evaluating network performance, rather than focusing solely on the performance of a single server. The results of these tests can then be compared to evaluate the performance of the shared hosting system.

Chapter 7

Synthesis

7.1 Results

This thesis demonstrates that the assessment of web hosting performance in the era of hardware virtualization demands a systematic and comprehensive approach to accurately evaluate and compare the performance of different web hosting solutions. A thorough understanding of the various factors and underlying constraints that impact web performance is essential to effectively measure website performance. This necessitates a comprehensive investigation of the various components that contribute to web performance, including hardware resources, network infrastructure, and software configurations.

7.1.1 The Need for Specialised Benchmark

In general, *general benchmarks* are not suitable for measuring the performance of web hosting systems, as they are typically designed to evaluate desktop performance and not server performance. A comprehensive and diverse approach is necessary to accurately evaluate the performance of a web hosting solution, taking into consideration various factors such as CPU utilization, memory utilization, network infrastructure, and storage utilization. Utilizing benchmarking tools that are not specifically optimized for server environments may not provide the level of detail and specificity required to effectively evaluate the performance of a server. **It is therefore important to be cognizant of the differences between benchmarking tools optimized for desktop and those optimized for server environments, and to choose the tool that is best suited for the specific requirements of the environment being evaluated.**

The use of *Web Server Benchmark Utilities* is prevalent among DevOps personnel and web hosting experts due to their user-friendly interface, comprehensive analysis of resource utilization and web server performance, and the availability of a quick installation process and command-line interface, which is favored by many developers working in Unix or Windows server environments. However, in limited environments where only PHP and MySQL functions can be executed, the implementation of

such utilities may not be feasible. **As a result, traditional web server utilities cannot be utilized in the benchmarking approach adopted in this study.**

Self-hosted HTTP Load-testing Tools may not provide an accurate representation of real-world performance as viewed from an external perspective. Although these tools can be useful for server administrators and DevOps professionals in optimizing system performance, they do not account for limitations imposed by web application firewalls, load balancing, or other security measures implemented by hosting systems. These external factors can also influence the measured performance. Additionally, the simultaneous execution of hundreds or thousands of requests from a single client is not a common scenario. **Consequently, this method of performance measurement may not be capable of adequately simulating a genuine surge of visitors and, thus, yielding meaningful results.**

7.1.2 PHP Benchmarking

The utilization of *Pure PHP Benchmark scripts* as a means of evaluating the performance of a web host is limited in its scope, as it only measures the performance of various functions and operations in PHP, and does not account for other factors such as memory utilization, file system and database access. It is therefore concluded that a pure PHP performance measurement is merely a component of a comprehensive benchmarking suite.

Similarly, relying solely on *Synthetic Benchmark Tests* may not yield a comprehensive understanding of the performance of a web host, as the results obtained may not accurately reflect the various real-world factors that impact the user experience. While the data obtained through these tests is important and can be included in a later evaluation, it is insufficient to make meaningful conclusions about the performance of the host.

To obtain a more accurate assessment of a web host's capabilities, it is necessary to consider multiple factors, including memory utilization, file system, and database access, and to test the host's performance under real-world conditions, such as running a content management system or other complex applications.

7.1.3 WordPress Benchmarking Plugins

Specialized Content Management System (CMS) plugins can be a useful tool in evaluating the suitability of a web hosting server for a particular website and provide valuable insights into its performance capabilities. However, they may not be suitable for a generalized web hosting benchmarking suite, particularly with an eye towards potential future commercial implementation. **Therefore, it is advisable to consider conducting synthetic tests in an environment that is independent of a specific CMS, so that the results can be applied to a wider range of applications, including custom developments and those utilizing other CMSs.**

To make accurate and reliable assessments of server performance for a specific WordPress project, it is imperative to understand the impact of various factors on benchmarking results. The valuable insights gained from exploring the effects of different configurations of WordPress on server performance highlights the importance of avoiding non-standardized WordPress instances for conducting synthetic benchmarks. **Thus, when utilizing WordPress for benchmarking, it is crucial to take into account the potential influences on the results.**

7.1.4 External Benchmarking

External benchmarking can serve as a valuable method for assessing the performance of a web server or website. However, its reliability can be diminished when the server or website is only accessible through a Content Delivery Network (CDN). This is because the performance of a CDN is dependent not only on the performance of the origin server, but also on various other factors such as the network topology, caching policies, and load balancing algorithms. Measuring common metrics, such as response time, requests per second, page speed, and up-time, using external load testing tools may also not be feasible due to the widespread use of CDNs and server-side caches [72]. *Nevertheless, these metrics can provide valuable information to customers, and therefore should be included in a comprehensive evaluation of web hosting performance.*

Website Speed Testing Tools, widely utilized in both academic research and website development and optimization (see Table 6.1), provide metrics that website operators are already familiar with and can use to compare with the performance of their own website. *As a result, commonly recognized values from such tools should be part of the benchmarking suite and factored into the overall ranking.*

The field data from **Google Pagespeed Insights** is generally not available for artificial websites built specifically for testing purposes. *Therefore, this tool cannot be included in a benchmarking suite.*

WebPageTest, on the other hand, is a widely used tool for evaluating the performance of web pages. It offers a variety of detailed "Page-level Metrics" that provide insight into various aspects of website performance, such as page load speed, data transfer, and user experience[70]. *For the purpose of this thesis, the most relevant metrics for evaluating the performance of a web host are considered to be Time to First Byte (TTFB), load time (also known as fully loaded), bytes in, chromeUserTiming.LargestContentfulPaint, and chromeUserTiming.firstContentfulPaint.*

Online load-testing tools provide a viable method for performance testing by simulating concurrent users or requests to the system under normal and peak conditions. The objective of load testing is to evaluate the system's performance by subjecting it to a large number of users or requests under various conditions. *However, to accurately determine the upper limit of the number of visits that can be processed stably, it is essential to push the web server to its limits and beyond until requests fail.* This is because it is only

by examining the system's behavior under extreme conditions that its performance limits can be accurately established. If the system does not fail during the load test, it indicates that the limit of stably processable visits has not yet been reached. *Hence, it is crucial to continuously increase the load until the system fails to determine the actual limit of stably processable visits.*

7.2 Comprehensive Proposal for Problem Solving

As this thesis shows, no single benchmarking approach is universally applicable or perfect. Each benchmarking methodology has its own strengths and limitations. But, by identifying the specific requirements and unique characteristics of different sub-areas, it is possible to create a more comprehensive and meaningful benchmark by combining and balancing the individual metrics obtained from various benchmarking techniques.

7.2.1 A Multi-Layered Approach

It was found that each of the analysed types of benchmark has its own strengths and weaknesses; therefore, a weighted evaluation of different benchmark data for different use cases is essential, resulting in the development of a multi-layered approach to benchmarking WordPress-specific hosting environments. In order to ensure that the performance evaluation is independent of the underlying operating environment and implementation, a holistic approach has been adopted in earlier research [8]. A general approach assumes that a consistent methodology is applied to all components of the workload. Additionally, an approach for integrating these individual metrics into a comprehensive metric for comparing systems has been presented. However, it is important to note that the assessment and forecasting of performance are influenced by a wide range of user requirements. Therefore, there is no definitive answer for which workloads should be included and how they should be integrated. Consequently, this high-level approach allows for flexibility in accommodating different workloads and integration methods that align with specific user requirements.

In a study by Casazza, a high-level approach was adopted to avoid dependence on specific operating environments and implementations. The study presented an approach for aggregating workload components into a unified metric for system comparisons, taking into account the coherence of performance discipline on each component [8]. Casazza also acknowledges that there are numerous user requirements for evaluating and projecting performance, and as a result, there is no universally accepted answer for what workloads should be considered and how they should be combined [8]. **In accordance with this research, a two-fold approach is proposed for evaluating the performance of web hosting solutions: Both synthetic benchmarks in a controlled laboratory setting and simulations of everyday conditions are suggested as means of evaluation. This approach will offer a more comprehensive and**

representative understanding of the performance characteristics of web hosting solutions.

To further validate the scientific validity of this approach and facilitate marketing efforts by a web hosting comparison portal, potential weighting keys can be considered for different target groups of web hosting customers.

7.2.2 Components of a Benchmarking Framework

This thesis endeavors to advance a potential implementation of a benchmark for assessing web hosting performance within the context of hardware virtualization. To achieve this objective, a multi-faceted benchmark approach is proposed that utilizes granular Web Hosting Performance Metrics, as follows:

It has shown, that each approach has its own strengths and limitations. While synthetic tests strive for minimal falsification in order to ensure comparability and reproducibility, real-world testing, aims to conduct testing under everyday conditions in order to obtain results that are most reflective of the experiences of actual users.

Therefore, a combination of synthetic tests and real-world testing is suggested to provide a comprehensive and balanced evaluation of a web hosting service's performance.

7.2.3 Frontend and Backend Performance Testing

The real-world performance evaluation places emphasis on both the user experience of the website and the backend performance as perceived by the website owner. In order to obtain accurate and meaningful results, it is preferable to use a standardized WordPress instance for measuring real-world use cases rather than synthetic tests. This approach provides a more comprehensive evaluation of the performance of the system, as it accounts for the various factors that may affect the user experience.

The utilization of WebPageTest.org is suggested in order to assess the front-end performance of a website. The proposed methodology involves running an EC2 instance from multiple geographical locations in order to account for the effects of Content Delivery Networks (CDN) and caching. It is noted that front-end performance is a critical factor for customers, as it influences their experience and often serves as a point of comparison between different websites. By utilizing WebPageTest.org with the aforementioned settings, it is believed that a comprehensive evaluation of the website's front-end performance can be obtained.

Additionally, a method for evaluating the backend performance of a website, by measuring the runtime or response time of the "wp_admin_ajax.php" file in a standardized, default WordPress instance is suggested. This approach is designed to provide an unbiased look at the performance of uncached requests, thereby serving as a baseline for assessing the backend performance of the website.

It is noted that additional measurement methods may be required in order to accurately evaluate the impact of acceleration methods such as OpCache, database cache, Redis, and others. These methods may include measuring the response time of specific requests or analyzing the performance impact of different caching configurations. The goal of these additional measurements is to gain a comprehensive understanding of the website's backend performance, including the effects of acceleration methods, and to identify areas for improvement.

7.2.4 Synthetic Performance Testing

The use of synthetic testing enables the collection of data in a controlled and standardized measurement environment. The metrics utilized, as listed in Table 6.2, encompass various aspects of performance, including CPU performance, disk performance, MySQL performance, and network performance. The CPU performance, for instance, is measured by the Pi measurement, which records the time taken to calculate the mathematical constant pi in milliseconds. Additionally, the disk performance metrics include measurements of read, write, and compile operations, expressed in MegaBytes per second. The MySQL performance metrics encompass various database operations, such as connecting, inserting, selecting, searching, updating, and deleting, and are measured in terms of the number of rows processed per second. Lastly, the network performance metric is represented by the Fsockopen measurement, which records the upstream and downstream data transfer rates in Megabits per second.

These granular metrics provide a comprehensive evaluation of the underlying performance capabilities of a server across relevant domains, and offer a practical option for assessing web hosting performance within the context of hardware virtualization.

7.2.5 Stress Testing

An internal benchmark, by its intrinsic nature, lacks the capability to determine if it is functioning within the parameters of a soft limit when executed. This presents implications on the performance of web hosting systems, as it results in incoming requests being queued and processed only when a PHP worker becomes available. Thus, it is not feasible to accurately determine if the system is operating optimally in terms of responsiveness. It is only upon reaching the hard limit of the system that this is indicated by a server unavailability message. **Therefore, the use of an external load testing tools such as k6 is crucial in the benchmarking process of web hosting performance.**

Stress testing is a specialized form of load testing that focuses on determining the limits of a system and its ability to handle extreme conditions. Stress testing is essential in determining the limits in terms of concurrent connections and the number

of PHP threads and PHP workers. Through load testing, the capacity of a web hosting system can be assessed, allowing for meaningful results to be obtained in the benchmarking process. During stress testing, two distinct limits can be identified - the soft limit and the hard limit. The soft limit is indicated by delays in performance, while the hard limit is indicated by server errors 5xx. It is important to note that when approaching these limits, the stability of the web hosting system should not be endangered. Thus, a gradual and systematic approach should be taken in testing to ensure the reliability of the results obtained.

In order to ensure that the benchmark results are accurate and meaningful, it is important to ensure that the benchmark is properly configured and that the test scenarios closely mimic the real-world usage of the web hosting system. Additionally, it is important to monitor the performance of the system under test in order to gain a more comprehensive understanding of its behavior and responsiveness.

In conclusion, the use of load testing tools such as k6 is imperative in obtaining meaningful results in the benchmarking of web hosting performance. Through stress testing, the limits of concurrent connections, PHP threads, and PHP workers can be determined, providing a comprehensive assessment of the system's capacity. However, caution should be taken to ensure the stability of the system during testing.

Note: When conducting a stress test, it is important to obtain proper permissions and approvals to avoid misinterpretation as a denial of service attack. Clear communication should also be established to prevent any potential misunderstandings or disruptions during the testing process.

It is also advisable to conduct multiple iterations of load testing in order to establish a comprehensive database. This is because given web server configuration may demonstrate adequate performance under certain traffic loads, while under other workloads its performance may not be sufficient.

7.2.6 Rolling Benchmarking

The evaluation, projection, and reconfiguration of web systems can be challenging due to the complexity arising from various factors such as hardware specifications, software characteristics, multi-layer architecture, and network-related aspects that may change over time. As a result, it is advisable to conduct multiple tests in order to obtain a more comprehensive data set.

To ensure optimal performance of web hosting environments, it is recommended to continuously monitor and benchmark the system, which will enable the identification of performance limitations resulting from concurrent background processes and facilitate the assessment of performance trends over time.

7.2.7 Prudent Implementation

In summary, it is imperative to consider the following factors when carrying out any of the benchmarking methods discussed in this thesis:

1. It is of utmost importance to take into account the unique characteristics of the system and the functions being used to accurately capture all relevant variables.
2. After implementation, rigorous testing must be performed on established hardware in various hosting environments, serving as a reference for comparison of the obtained results and reducing the likelihood of unexpected behavior.

In view of these considerations, strict adherence to the benchmark requirements outlined in Section 4.6.6 by Huppler is required. By adhering to these guidelines, the outcomes of the benchmarking process will be accurate and meaningful.

Chapter 8

Discussion

The discussion presented in this chapter aims to contribute to the broader scholarly conversation within the field of web science, as well as provide practical guidance for practitioners seeking to apply the findings in their own work. The chapter also evaluates the validity, reliability, and limitations of the methodology and techniques employed throughout the study.

8.1 Limitations

In this thesis, several influencing factors were taken into account, however, there are still numerous considerations that must be addressed prior to the achievement of a stable implementation. In practice, the web hosting system being measured can be any complex architecture, such as a derivative of this AWS reference implementation of WordPress hosting on Amazon's web services infrastructure (See Fig. 8.1).

It is also crucial to acknowledge that the evaluation of a hosting system goes beyond its technical capabilities, as it is also influenced by non-technical factors such as availability, security, and customer support. Thus, it is imperative to take these factors into account when conducting an evaluation of a shared hosting system to ensure a comprehensive and well-rounded assessment.

8.1.1 Unclear System State

It is imperative to take into consideration the presence of resource-intensive processes during performance measurement, as these processes can have a significant impact on the results. In the absence of Command Line Interface (CLI) tools such as `top`, `VmStat`, `htop`, or `iostat`, customers and users of hosting services may not be able to accurately identify the presence of such processes, such as backups or system updates, running in the background. The utilization of these tools is crucial in order to accurately assess the performance of the hosting services and minimize the potential impact of background processes on the measurement results.

As these tools are not available in in managed hosting and shared hosting environments, it is necessary to take these possible influences into account during

Editor's note: the figure was removed due to copyright issues.

FIGURE 8.1: WordPress Hosting – How to run WordPress on AWS.
Source: Cutout from [45]

each measurement. Furthermore, the evaluation of data obtained from multiple measurements conducted over time is necessary in order to gain a comprehensive understanding of the performance of the hosting services. This approach will provide a more robust evaluation of the performance and help to account for any fluctuations or anomalies that may occur.

8.1.2 Benchmark Variation

Website speed testing results can vary due to factors such as service, location, browser, and time of day, like Bartuskova showed [37, p. 133]. Therefore it is recommended to use consistent settings, running multiple tests, and excluding significant deviations for more accurate results. But further research is needed on the potential impact of time of day on website speed testing, according to the author [37, p. 133].

It is important to note that the results of website speed tests can be influenced by factors like server load, equipment, and time of day, and should be treated as rough estimates rather than definitive indicators of shared hosting performance. Averaging results from multiple tests may improve accuracy.

8.1.3 Benchmark Manipulation

It is theoretically possible for system engineers to detect when a benchmark is being run on a system and manipulate the observations of the benchmark. For example,

by detecting the presence of the benchmark and activating special performance optimization techniques. System engineers may include code in the system that detects when a benchmark is being run and activates specific optimization techniques that are designed to improve the performance of the system under the benchmark. These optimization techniques may not be activated under normal operating conditions, as they may not provide any benefits or may even degrade performance.

While none of the interviewees mentioned, or admitted to such behavior, it is possible that benchmarking could be deliberately thwarted or manipulated.

Overall, it is important for benchmark designers and users to be aware of these potential issues and to take steps to mitigate them, such as using techniques that are resistant to manipulation, using multiple benchmarks to validate the results, and using benchmarks that are representative of the intended workloads of the system.

8.1.4 Validation and Limited Testing

It is imperative to put into practice the methodologies discussed in this Master Thesis through coding and testing them on real hardware and hosting environments for their validity. However, the implementation of these approaches in code would entail substantial time and resource allocation, which were not attainable within the confines of this thesis. Furthermore, conducting the tests on real hardware and hosting environments would necessitate access to these resources. While it would be ideal to validate the approaches through implementation and testing, the constraints on time and resources make this infeasible.

As a result, it is necessary to rely on theoretical analysis, existing benchmarks and empirical results from the interviews in order to evaluate the effectiveness and feasibility of the proposed approaches.

8.2 Further Research

The next step for a further research would be the development of a modified version of the benchmark, testing and validating it to ensure it meets the objectives of the thesis. Testing the approaches on real hardware and hosting environments will be challenging due to the complexity and variability of these systems. This requires extensive testing in order to ensure that the approaches are effective and reliable.

Based on the analysis of the benchmark code and the identification of limitations and areas for improvement, a modified version of the benchmark should be developed to suit the objectives of the thesis. This includes making any necessary changes to the benchmark's methodology, the types of tests it performs, or the specific PHP and MySQL functions that are used.

The next step is to test the modified benchmark to ensure that it is suitable for use in a limited (shared) web hosting environment and that it meets the requirements of the benchmark. This includes testing the benchmark on different web hosting environments and comparing the results.

8.2.1 Edge Computing as a future technology

Edge computing is a distributed computing paradigm that brings computational power and data storage closer to the devices and users that need it. By moving computation and data storage away from centralized data centers and into the network edge, edge computing can reduce latency, increase data privacy and security, and improve the reliability and scalability of web hosting environments.

For these reasons, edge computing may have a significant impact on the web hosting environments in the future and makes it an important topic for future research projects that aim to understand and optimize the performance and scalability of web hosting environments to provide a more accurate and realistic representation of the performance and user experience of web applications and services.

8.2.2 Adjusting Metrics for Benchmarking

When conducting a comprehensive evaluation of hosting companies and offerings, it is imperative to consider the different segments of customers and target groups present in the hosting market. These groups range from individuals embarking on the creation of their first website to established medium-sized corporate websites, e-commerce businesses, and multinational corporations operating on a global scale. In light of this diversity among target groups, it can be posited that each group has unique priorities when evaluating a hosting provider. As a result, it is suggested that the benchmark suite should only present raw data and the interpretation of this data into a score or ranking should be performed only after a thorough investigation of the target groups, their requirements, objections, apprehensions, and concerns. Consequently, the weighting of metrics should be adjusted based on the needs and preferences of the target audience. For instance, in the context of shared hosting, the cost of the hosting package and front-end performance may be more vital than back-end performance, whereas for a Virtual Server offering, customers may attach greater importance to PHP performance and network performance.

This raises the significance of further research, that should focus on understanding the interpretation of benchmark results, the metrics employed, and the appropriate and effective visualization of the data. Through this, the benchmark suite can better cater to the needs of each target group and provide more informative results.

8.2.3 Further Relevant Metrics

This thesis also suggests that for a comprehensive evaluation of web hosting performance, it is recommended to consider a balanced combination of various metrics, such as those listed in the synthesis in Chapter 7, and factors such as service quality and compliance. This combination of values helps to provide a basis for comparison and increased transparency in the measurement of web hosting performance.

8.2.4 Further Research Questions

The following research questions emerge from the proposed benchmarking methodology:

1. How can the proposed benchmarking methodology be executed, validated, and calibrated effectively?
2. How can market research and user surveys be used to gather information on the requirements of different target groups in web hosting?
3. How can the metrics be scaled, evaluated, transformed into grades, and weighted appropriately?
4. How can the collected data be processed, compared, and presented in a clear and compelling manner?
5. What additional factors such as compliance, environmental impact, and social considerations should be taken into account in the evaluation process?
6. How can user-experience testing of hosting management interfaces be integrated into the evaluation of web hosting solutions?
7. How can the methodology be adapted for use in diverse web hosting environments and extended to include cloud-based solutions?
8. What techniques are available for detecting, manipulating, or interrupting the benchmark execution, and what are their limitations?

It is important to consider that the above enumerated questions represent merely a few potential avenues for further research that have the potential to mitigate the limitations of the proposed benchmarking methodology and furnish more comprehensive and precise assessments of web hosting performance.

Chapter 9

Conclusion

In conclusion, the evaluation of web hosting performance in the era of hardware virtualization requires a comprehensive and systematic approach. General benchmarks are inadequate for measuring web hosting performance, and it is crucial to choose a benchmarking tool specifically optimized for server environments. Utilizing web server benchmark utilities is common among experts, but it may not be feasible in limited environments. Synthetic benchmark tests and PHP benchmark scripts alone may not provide a comprehensive understanding of web hosting performance. It is important to consider multiple factors, including memory utilization, file system, and database access, and to test the host's performance under real-world conditions. Specialized CMS plugins can be useful in evaluating web hosting server performance, but it is important to understand the impact of various factors on the benchmarking results when using WordPress. External benchmarking can provide valuable insights, but its reliability may be affected by the use of a CDN. Website speed testing tools and online load-testing tools offer valuable metrics for evaluating web hosting performance and should be included in the benchmarking suite. To accurately establish the performance limits, it is crucial to push the server to its limits until requests fail.

A multi-layered approach to benchmarking has been proposed, which involves a combination of different metrics obtained from various benchmarking techniques (see Table 9.1), taking into account the specific requirements and unique characteristics of different sub-areas.

For real-world testing, a standardized WordPress instance is used to evaluate both front-end and backend performance. The front-end performance is assessed using WebPageTest (See Figure 9.1). The backend performance is evaluated by measuring the runtime of a non-cached backend script. Additional synthetic tests allow the collection of data in a controlled environment and uses granular metrics to evaluate the performance of CPU, disk, MySQL, and network performance. Finally, stress testing, using an external load testing tool like k6, is crucial in determining the limits of a system's ability to handle extreme conditions (Load Testing Cloud in Figure 9.1).

This approach offers a comprehensive and representative understanding of the performance characteristics of web hosting solutions. Additionally, potential weighting

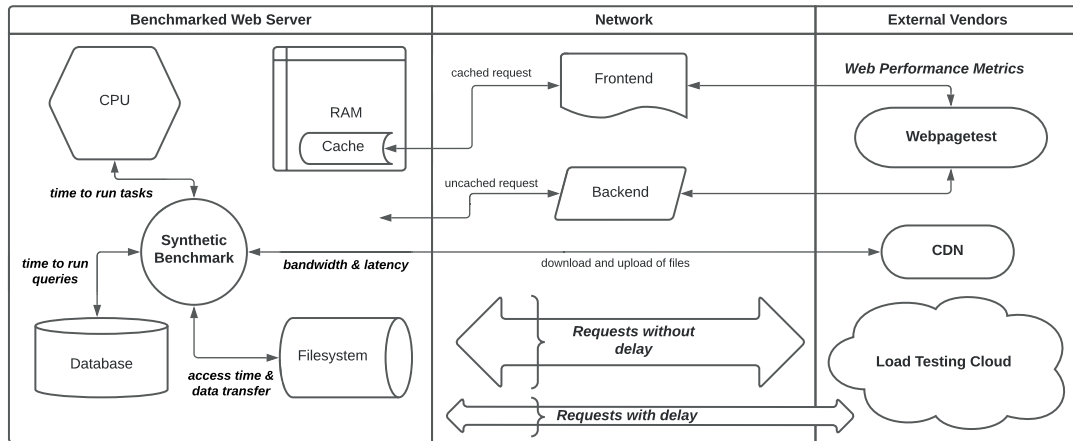


FIGURE 9.1: Multi-layered approach to benchmarking web hosting performance. Source: Own Figure.

Category	Description	Metric(s)
Real-world testing (Section 7.2.3)	Front-end performance assessed using WebPageTest.org, Backend performance evaluated by measuring runtime of non-cached backend script in a standardized WordPress instance.	Web Performance KPIs: TTFB, loadTime, bytesIn, LargestContentfulPaint, firstContentfulPaint
Synthetic testing (Section 7.2.4)	Collects data in a controlled environment. Granular metrics to evaluate CPU, disk, MySQL, and network performance.	CPU: time to run tasks, Disk: access time and data transfer rate, Database: time to run queries, Network: bandwidth and latency
Stress testing (Section 7.2.5)	Determining the limits of the system's ability to respond to parallel requests. Using external load testing tool.	Maximum number of parallel requests, without noticeable delay.

TABLE 9.1: Overview of the multi-layered benchmark approach

keys for different target groups of web hosting customers can be considered to validate the scientific validity of this approach and facilitate marketing efforts by a web hosting comparison portal.

Appendix A

Interview Transcripts

In this thesis, the transcripts of the interviews conducted were anonymized to protect the confidentiality and privacy of the participants. This means that all personally identifiable information such as names and contact information was removed from the transcripts. In addition, to prevent possible inferences about the identity of the interviewees, measures were taken to obscure or eliminate certain details that could serve as identifiers, such as naming companies or other identifying information in the transcripts.

A.1 Interviewee 1: Pat

Interviewer: Kai Priestersbach (K)

Interviewee: Sole Proprietor (P)

Date and time: December 9th 2022 17:34

Location: Virtual meeting via ZOOM

K: We are now in a completely virtualised world. I think hardly any hoster still operates something like a real root server or something. Even if you have your own server, they are also virtual resources in a larger environment. And there is the question: What possibilities are there for allocating resources? And that's why the guideline now is more or less like this: What does it look like for you when you set up software or hosting, what does the stack look like? So which virtualisation solution, which customer management, which web server? Above all, the topic that is super exciting is how resources are managed, monitored and allocated. What limits come into play? Then a bit of general information. What are the most frequent customer requirements? Which CMS do you use? Why do clients choose you or you? What do you do when a client complains about, I don't know, the backend is too slow, the website is too slow. How would your systems react if there was a short-term rush of customers? How do you distinguish an access rush from a DDoS attack? Which, exactly, and which benchmarking functions do you know or use to optimise for this? Either to see for yourself how fast your stuff is that you're building or actually

for, yes, the classic is PageSpeed. So most hosters try to optimise the Google PageSpeed somehow so that it looks nice from a marketing point of view. So there/

P: Do you mean with the hoster of the website itself?

K: Yes, the hoster on the website or the hoster for the websites of his customers. That they don't optimise for a single customer, but rather build up their systems in such a way that, for example, NGINX activates a full-page cache or something. Something like that. So I would simply start at the top. Just guide me roughly through the stack from the hardware layer, so to speak, to the finished customer account.

P: So for us it's actually like that, we've been dealing with hosting virtual machines for I think 14 or 15 years. And with the special customers, where I was also a permanent employee at that time, we set up a completely new cloud system every two years, and we didn't host the web, but it was about keeping the Microsoft Cloud running, although each customer had their own cloud. Yes. And that's another thing. If the performance is not right, then it's almost like a web server, then a complete server crashes very quickly. And that's why we dealt with it quite a lot, with hardware etc. and back and forth. But I have to say quite honestly that we have two systems running right now. One is actually a root server, which means that the cloud system is distributed over several root servers and more or less scales itself. And the newer system, we are currently playing with this [HOSTING PROVIDER] Cloud.

K: Ah okay.

P: And this has been the case since day 1 of the [HOSTING PROVIDER] Cloud, because I had a very good contact with a [HOSTING PROVIDER] technician with whom we had also optimised our own systems. So that we are guaranteed to have the performance we need. And above all, there is the crux of the matter, but it's a whole separate topic: how do you actually measure the performance of a cloud that runs via [HOSTING PROVIDER]? Because this is a very complex issue, because the things that are displayed in the cloud have nothing to do with the server. And if one cloud then pulls down the other clouds, then you have no CPU utilisation in this cloud. But it's still really slow and things like that.

K: Yes, in fact the topic of cloud hosting, which I then want to solve second next year after the topic of WordPress hosting, is that you miss that neatly. But that's not really part of the master's thesis, or at least only in the sense that I'm comparing it. I'm just saying that one model is classic. You keep a certain number of resources and distribute them among all the customers and hope that one customer doesn't drag down all the others and that everything balances out

a bit. And the other thing is that the systems scale completely in a very large environment and billing is really dependent on consumption.

P: Yeah, well, I can go to/

K: [unintelligible] other approaches, stop.

P: Yes, exactly. I would now simply like to talk about this use of hardware, which ultimately ended up in our final and is probably also simply state of the art, I would say. Because no one has much leeway. So everyone is always bound to the hardware and in the end there is always a box behind it. Because you just mentioned, not the classic hosting. For me it still feels like that/

K: Yes, at the end of the day, of course there are servers in the rack, sure.

P: Yes, it still feels that way to me. But for us it's really like that, we ended up with one product and that's the keyword RMM. You've probably heard that before. Remote monitoring and management.

K: I see. Yes.

P: And that is actually the crux of the matter. And I wouldn't even go into the hardware so blatantly right now. But as a basis, you can imagine, you have different types of servers or Notes. That is, you have a server that only provides CPU power, for example. Or several. Then a server system that provides the hard disks and/ or the CPU.

K: That is quite typical. A database system, a storage system, a computer system, so to speak.

P: Yes, the funny thing is that we have tried to make the virtualisation as detached as possible. That means you can use several/ This is extremely fast and really good. You have one box and use the network/ In this case, we chose fibre optics between the servers. So the CPU is/ You can use the CPU from one server from other servers.

K: Okay, nice.

P: With VMware.

K: The latencies are not a problem?

P: Yes, so the latencies are surprisingly not a problem there. And this construct looks like this. That's why we have in this system, I don't want to name the customer now, because/

K: Yeah, no, don't do that. Oh, that's right, just for your information. It will be anonymised. So in the published work, it doesn't say your name or the company or anything. It just says, I don't know, hosting expert of a blah blah blah.

- P: Well, I'm not giving away any big secrets, at least I don't think so. It's just the empirical values. So we would have/ In this last system, which we had just set up with them, there was just once the computer where the VM runs on it.
- K: This is VMware based, so to speak?
- P: Exactly, VMware-based. So the RMM, this management software, we used/ What was it called? N-able, so N-able is what it's called. It's a paid software.
- K: All right, these are all in the enterprise sector.
- P: Yes.
- K: You want it in that area because of support etc., don't you?
- P: Yes, they have a really good team behind them and they also provide support very quickly when something goes wrong. Because in the sector, if something goes wrong, you simply need fast support from this software. Whereby it's actually like an administration software. So/
- K: Yeah, I'm on the website right now. Okay.
- P: So you have to think of it like this. You have a server that runs on it. The VM is quite banal as you know it. Then you have a hard disk in this VM that runs on the server itself. An operating system runs on it, so to speak.
- K: And the image? Exactly.
- P: And then we have a file server. That is actually a server. I think the last time there were 16 hard drives or so with Raid 1 inside and the two servers are connected via a switch, so with fibre optics. And then we have a server that can provide memory and a server that provides RAM.
- K: Oh really? That means that the data is fetched from the hard disk via the network, executed in the CPU on one side and stored in RAM on the other. That's weird.
- P: And/ So we have had really good experiences with it. The whole thing works like this. You always have two nodes so that the whole thing runs redundantly. And now comes the crazy part of the whole thing. You can migrate the VM between two servers during runtime, while the VM is running. And that is/ was already possible about eight years ago.
- K: Crass.
- P: Now this is not a new, new idea or anything.
- K: Is this a standard functionality that VMware provides?
- P: Yes.

- K: Wow. Ok. I only know it from the Proxmox world. It works similarly there. But/ That you separate the CPU, RAM and hard disk or storage on dedicated resources.
- P: No, it's not like that. It's not like that. You have to imagine it like this. It's not like the entire CPU is pulled from the other server, but the VM runs autonomously and you can now switch on the size.
- K: But is it just the services or is it actually the execution? When you say, I have a rush of visitors now and I need more, so to speak.
- P: Yes.
- K: [unintelligible] I'm going to say this stupidly.
- P: That is also [unintelligible]. That's for the service, that's for the permanent operation. And the joke of it is VMware monitors all these/ They have some kind of artificial intelligence inside. It monitors the processes of the system. And you always have, that's quite normal in the CPU and the RAM, certain errors that occur due to software and VMware can recognise them in advance and then directly as error messages to accelerate the system.
- K: How in advance?
- P: So first/ Before the CPU validates and corrects this error, so to speak, the VMware just shuts down already and says, error in the query into the CPU. It's quite funny.
- K: What is the name of the VMware product?
- P: Um, I actually have to/
- K: Probably this ESX, right? ES/, this hypervisor?
- P: Oh, right. This is the hypervisor.
- K: Then it is probably the ESXi.
- P: Yes.
- K: Okay. But that means that at the end of the day you no longer/ How do you even know/ So if a customer comes in, I don't know, he says he somehow only wants to spend 10€ and you have to build him a small package.
- P: In the construct, it doesn't work at all.
- K: That's not possible, is it? Then how do you know how to use resources?
- P: No, you can say this VM has two CPUs and 32 GIG RAM/
- K: Guaranteed. So they are also kept permanently free?

- P: These are guaranteed to be kept free or you can also define shared CPUs. Then you say that the VM may use a maximum of 30% of the CPUs and then it becomes hard between the systems.
- K: Yes, exactly, that's probably in a mass hoster. So if you were to really build a shared hoster like that....
- P: Yes, if you were to ask [HOSTING PROVIDER], they also have this variant of the shared CPU or dedicated CPU. And that is exactly the same system. Funnily enough, [HOSTING PROVIDER] has actually built almost exactly the same system as we built in the data centre eight years ago or so. That's why I don't see all this as Rocket Science, but rather as standard, I would say.
- K: Yes, it's not. Honestly, by now the whole DevOPs thing is too, no one does their own thing any more. So that's also normal.
- P: And the next point is that when this system is running, there is a fallback system and then a backup system. And there it is again, VMware has a funny backup system that does real-time backups during real time.
- K: Real-time replication too, right?
- P: Yes, and we also have that in use, because especially with these customers or with the hoster, it is extremely important that the current processes are backed up, because a lot of data is calculated, [unintelligible] This VM system.
- K: Okay.
- P: And if there's a power cut or no idea what, it would be really fatal if they had to produce data for an hour.
- K: Okay. Yes good but/
- P: And/
- K: But that's actually the case with every major e-commerce retailer. If you have a steady stream of bookings and you lose an hour, and that's, I don't know, 15,000 orders gone. I think the requirement is the same for everyone who somehow has a certain volume, yes, transaction volume, no matter in which area.
- P: Yes, that is/ Oh God, if I had to imagine now that I would have to produce such a server quickly with a total failure, then I can actually pack up. Exactly.
- K: But these are not really high-availability applications that are running there. It is actually only one/
- P: With this client actually yes.
- K: Yes. Okay.

- P: So it's like this. I can describe it to you roughly. There are a lot of HR departments that actually run their entire software on it, complete documentation management and accounting and everything that has to do with it somehow. And one critical thing is time recording. So they have 300 customers and/or it's anonymous, right? They have 300 customers and almost every one of these customers uses the time recording and it all runs together in this VM per customer.
- K: Okay, yes, you can't tell everyone everything, the data got lost today. Look up your times.
- P: Yes, so if the system fails at night on New Year's Eve, let's say at twelve o'clock on New Year's Eve, and people book themselves out, then drama is inevitable.
- K: Okay. But that means there/ Yes good/
- P: Because that was actually a case that we had there.
- K: Ah, ok. These are not really things to consider, so I'm not really interested in hot standby or real high-availability. For me, it would be much more exciting if you said that you can assign, I don't know, individual CPU cores, whether shared or dedicated.
- P: Then I want to elaborate a bit here.
- K: Are they standardised or what? You can't say that a CPU is not/ So they are not the same, so to speak.
- P: So in the system, all CPUs are actually the same. So I mean, a CPU has, is not just a CPU, but has so and so many cores. So if you now put a computer that/ So it's a/ I think what do the computers have? I think the biggest computer has something like 250 CPUs or so. And then in the end it's more like a CPU. So there are actually three physical CPUs on the board and then each CPU has its cores and these cores can then be assigned to a VM.
- K: Yes, exactly, but these cores still have a certain, shall I say, bandwidth and directional frequency.
- P: Exactly.
- K: So I only know about conversations with um/ Which one was it? Uhh, the, oh exactly HostPress. Right. They had been relying a lot on AMD for a while, Threadripper mainly. And then they discovered that they had tested an i9 or something. It was clocked much higher in terms of megahertz, and that had a measurable effect on the actual/ Because you know, single threaded PHP, it really depends on the clock frequency, and it made a measurable difference for them. It's amazing. Then they completely replaced all their CPUs, so

everything bit by bit and now they really go to high clock speeds because it makes a noticeable difference.

P: Yes, that's actually interesting. So/ Unfortunately, I personally haven't had that much exposure to these CPUs. We had Intel Xeon CPUs from the beginning and we didn't really compare the throughput rate with other manufacturers.

K: In other words, when you say here is a dedicated core, a dedicated core or I don't know, the customer now gets four cores, these cores are actually reserved by this physical CPU.

P: Exactly.

K: So it's not going to be now/ Okay.

P: So you don't always get exactly the same one, but that's then a pool of/

K: Yes, it's somewhere. Exactly.

P: But you only have the maximum number of VMs, i.e. really a physical maximum number per VM that are there. The same applies to RAM and hard disk capacity.

K: Because that's the thing I was wondering too. I spoke to [COMPETITOR 1] on the phone yesterday. You probably know them too.

P: Yes.

K: And as a customer, you can even make your own resources available to [COMPETITOR 1] Cloud if you have spare capacity. And then it can happen that [COMPETITOR 1] runs some customer applications on your hardware and you get money for it, so to speak. And there/

P: It's actually a similar principle over the net, isn't it?

K: Yes, it's basically the same, except that they built it themselves. Or at least they say in which parts you really/ Yes, it doesn't matter. There are also of-the-shelf things like that.

P: So what I can say in any case is that we have rescaled this system every two years and in the tests that we ran/ So when we first set up a system where you could copy CPUs over the network and the file system over the network and then the VM back and forth in real time, we sat there and thought, how can that actually work?

K: Yes, that's what I'm asking myself at this point.

P: And the bandwidth is not that dramatically high. That is, if you now/

K: You don't need the bandwidth. It's all about latency.

- P: Exactly. Yes, that's the joke of it. If you now/ Of course we tested it with CPU load and so on and the bandwidth was surprisingly low. Of course, the latency is the decisive factor, but if you physically have a/ Well, we have/ They are plug-in cards with fibre optics that go into the router and there is actually practically no latency. Like this.
- K: Yes, that's true.
- P: Almost not measurable.
- K: Yes, that's right.
- P: And actually it's like a system, so a computer and we have/ With them it looks like this, you have racks and each rack is always a unit. And now they have two or four racks with this load. So with this configuration, as I described. And then a test system on which we carry out weekly maintenance or testing and then roll it out on the main system. That's also a big issue.
- K: Ah, ok.
- P: Because if you ever have a company where there are probably 30 servers or so. I haven't counted it now, but around that number. If at some point you have to go here and you have to update all these things, preferably during the runtime, then they/
- K: Then you want to know beforehand that it won't go wrong.
- P: It's going to be extremely exciting because VMwar is also changing, and a lot really changes over the years. And that's actually the main art for me in the whole thing, keeping the hardware running and the software running. So putting a cloud system here once, which is cool and which is fast, that's relatively easy, or at least with the knowledge we have, relatively easy. But keeping it alive is really an issue in itself, at least from my point of view, with the experiences we've had.
- K: But you don't go into the data centre yourself, do you? You then have remote hands when hard disks break down or how do you do that?
- P: That's actually/ We actually have a computer centre, which is in an old air-raid shelter at Münchner Freiheit.
- K: Oh, so the customer does that, so to speak?
- P: The customer hosts this himself.
- K: Oh, nice. Okay.
- P: That is why we have assembled the technology ourselves and have our own technicians. We have tried to copy this construct at two, among others at Host, at [HOSTING PROVIDER] and at another hosting provider, Munich. But they

didn't manage to set up the networks for us the way we wanted with the fibre optics, and they are/

K: Oh, as colocation or as [unintelligible]/

P: Yes, exactly, as a colocation.

K: They couldn't.

P: So that we can say we have remote hands in case of emergency. Because the customer is too small for it to be profitable to have technicians running around 24 hours a day.

K: Yes, that is clear.

P: And that means/

K: Damn it, another approach like that.

P: How do you mean approach?

K: But if you now have your/ You have a quite normal WordPress hosting also in the offer.

P: Yes, exactly. That's the one thing now. Yes.

K: That has nothing to do with it now, does it?

P: This has something to do with the conditional, because I actually built my first WordPress hosting from this system.

K: Okay.

P: That is, we have VMs with Debian Linux in this construct. We first started with CentOS and then realised that we didn't want to spend so much time on it, but rather wanted a system that simply had [unintelligible] packages. And then we switched to Debian and set up this system on top of Debian. And here's the funny thing, because we're lazy as hell/ You don't write that anywhere, do you?

K: No, it's called effectiveness and competitiveness.

P: We have automated everything. Completely. Really completely. In this construct, everything is simply completely automated with Ansible and Ansible recipes. And if we now say we need a, um/ We have a management software. In this case, it's ESP Config. And I have now also used ESP Config for the administration of the current cloud system.

K: For the customer accounts, that they then have their own login, so to speak, or what?

P: Exactly. That they have their FTPs, SSH and other things in the database exactly/ And we also had a great learning experience. We went here and first

set up, I think, four databases and four web servers. And then we moved the backups, all the dynamic data, that is, the www where the websites run, to the storage servers. And we then realised that the bottleneck was definitely not the database, but the web servers with these PHP processes and the NGINX that we had used.

K: Yes.

P: Or rather, we were still on two tracks. We had Apache and NGINX and now only NGINX remains.

K: Well, okay. Yes, it's also like that, because most of the websites are read-only. You don't have much of a database. Most of it can be cached. But if that's so separate, what about, I don't know, caching pages in RAM and stories like that? What you actually want to have with NGINX is also with, I don't know, OP cache and yes, you know, all the caching layers. These take place in RAM and the majority of the database is normally kept in RAM.

P: That's exactly why now, when now / So the web server where the websites run on, that's just our Nodeweb and technically PHP runs on it and the NGINX. And the database runs on the second server and the file system runs on a second server, which means that you have three servers for one website and / or the website itself.

K: So CPU and RAM are already on one.

P: Yes, exactly.

K: File and database is outsourced.

P: Yes.

K: Okay,

P: And the file server also has two layers. One is the backup system, which stores the data in real time, and then the file system itself. Which is then on SSD cards, i.e. NVME.

K: Exactly.

P: And the hard part is simple. We currently have six web servers, which you shouldn't say out loud, all of which have 32 GIG RAM and I think, oh God, I'm going to have to lie about that. I have to take a quick look.(...) How much CPU do we have? We actually do it like this. We make a new web server, then we start with 8 CPUs and 32 GIG RAM and depending on that / Then customers are added, customers are added. And that's already the new cloud that runs on our [HOSTING PROVIDER] infrastructure and we have /

K: Can you then simply book cores?

- P: Then you can/ exactly then you can book cores and memory up to a certain limit.
- K: Oh, OK. You basically just monitor the resource utilisation on your box and go up when necessary.
- P: Yes.
- K: And what are you looking at? So classic Linux command line tools, right? Or how can you monitor that cleanly?
- P: We monitor the entire network, old and new, with Zabbix. And there is my colleague, Peter, again, who/
- K: What is it called?
- P: Zabbix. So z a b b i x. I got to know this once at a conference where mainly hosters were there. And then they introduced us, they introduced me to it, because it was about monitoring systems redundantly and our Zabbix is also a redundant installation, because the problem is what happens if the monitoring fails. And I really think, I don't know, I bet you can look at every other hoster and they don't have redundant monitoring.
- K: Yes.
- P: And we just went here and built Zabbix like this, it's also open source. First of all, if a server fails, it is checked twice by one Zabbix server, which is located in Nuremberg, and the other one is somehow located in Helsinki at [HOSTING PROVIDER].
- K: Ah okay. You keep quasi/ That is already redundant and yes, so to speak, there is such a fail-over monitoring.
- P: Yes, exactly. Because it's really common that a monitoring system has problems monitoring something because of a network split or something else.
- K: Yes.
- P: And. Exactly. That means Zabbix monitors everything, it really monitors everything. So all the data that we could somehow grab from the system, we monitor with it. And Zabbix is such a system, you can just put everything in as you want and say, this is the data of the CPU, this is the data of the hard drives, this is the network card and so on. And Zabbix then turns these data into visual/
- K: Oh, you have different connectors, you put your data on them and then you can build dashboards or alerts etc. yourself.
- P: Exactly. Yes.

- K: Well, ok. There's also the, what's it called, Paessler. Paessler does something similar, I think?
- P: Paessler. Yes, exactly. Paessler is like a database, right? Or am I confusing the two?
- K: What is it called? PRTG? Exactly PRTG Network Monitoring from Paessler. I once did SEO for them, so I knew it in the background.
- P: No, I mean something else. It's one of those/ I can't think of it right now, I have to research it. There's a thing like that, you can throw everything into it. And then there's a certain artificial intelligence behind it and it can structure data and pick out anomalies.
- K: Okay, yes, good. There are several systems, depending on whether it's real-time or not and so on. That almost brings us to the topic of the data lake.
- P: Yes, in the end. But that means that if we now really look at how we measure the systems, we now have a construct built from this server, which is a virtual box that we can configure via share scripts or via the management software. Then on the one side we have the Ansible server, which also runs redundantly for us, which can then trigger these settings if we now say that one VM should get more CPU, that is.
- K: Do you automate this via Ansible scripts?
- P: Exactly, start the Ansible script. If in doubt, it shuts down the VM briefly, adds a CPU to the [HOSTING PROVIDER] API or whatever is possible at the moment or whatever is possible in these packages. It shuts down the CPU again and logs everything, and if everything goes well, then ideally we don't notice anything. And then we have the third part, the Zabbix server, which also has various alerts, as it does now, and an alert is not just triggered by an SMS or email or something, but Zabbix also triggers Ansible.
- K: Yes, it makes sense, if you suddenly need more resources at night, it runs from a Zabbix alert to an Ansible script and the resources go up or down.
- P: Exactly, that is already very customised what we have built there now.
- K: But why did you build it that way? Isn't there a ready-made solution?
- P: Yes, exactly. The finished solution is what I said at the beginning with the enable. The enable now monitors this hardware construct, which runs completely locally at the customer. And this thing can recognise when a CPU, when a VM suddenly has an extremely high load, and can even migrate it in real time to another server that has less CPU. That's really cool with the enable.
- K: Ah, nice.

- P: And the joke of it is, you have, because you always have the IP address problem and/or the issue is huge, really huge. You still have this special layer on top of the IP address and every single server has an IP address that has to be addressed. And if you migrate it from one server to another, you basically have a different IP address. And we have solved it this way, but the suggestion is/
- K: A central gateway or? Which has the same to the outside?
- P: Exactly. The enable is solved like this. You have an external and internal IP address. The principle is like that of a failover IP address.
- K: Ah yes, that's right. Yes, exactly.
- P: And you can also play around with it in such a way that you have a failover IP address that simply switches to another VM when one is busy. Or if the VM is now moved from one server to another, then it also switches the IP address directly after the migration. And it is solved in such a crazy way that if traffic still arrives on the old server, it is simply tunneled through to the new server.
- K: Oh, cool. Okay.
- P: This is the only way we basically do website removals.
- K: Yes, that's very cool, of course.
- P: We always tunnel the traffic.
- K: And especially if you still have people who still have the old one.
- P: Exactly you use/ Or when you migrate a shop/
- K: Now you have the replication problem again, when people arrive on the old one, but the new one is actually already running.
- P: Yes.
- K: But how does the tunneling work then? Well, that's not really relevant now, but I'm interested.
- P: It's totally simple. It's just an IP table value where you say, traffic coming in on the port where the IP is coming in will be forwarded to the/
- K: Oh, so that's rerouting actually in the network layer?
- P: Exactly, rerouting at the network level.
- K: Oh, how cool. Yeah, okay.
- P: And that's what I do with websites, moves and things like that.
- K: Yes, that's smart. Do you then also have a real hardware firewall somehow centrally? Or is it also something quasi theoretical in the same/.
- P: That's another layer on top.

- K: Already a layer of its own, right?
- P: That's one layer above. At the moment, we are still using IP-Fire as a firewall at the data centre, but to be honest, I'm not that happy. It's more of a legacy for them, I have to say in all honesty.
- K: Okay.
- P: But we have really beefed up the IP Fire. It also runs redundantly and has different network levels, red network, green network, DMZ and the DMZ is then this data centre. And there is really no way to get in or out in any form. And then they have a second network, which is called the black network. This is a network that is completely internal. And/
- K: Will you come there from the computer centre? Or not at all?
- P: You can actually only get there within the data centre. Or if you connect via a special VPN.
- K: Ah, okay.
- P: And that is the network, which is very critical from the security layer, because you can connect the hardware boxes there.
- K: Ahh.
- P: And the hardware network card is completely isolated from/ They have several network cards in there, actually three. Once the fibre optic thing, then the network card that gets the external traffic and then again this second network card that does an internal network, so the third network card that does an internal network.
- K: But that's actually how you want it. You want to have a clean separation as well. You want to have, for example, your management port, on which you install the operating system and so on, in the same network.
- P: Just a little fun fact. At the very beginning you could connect via ISDN/ There's even this ISDN port still, I think. You could connect via IS/ Because if all else failed/
- K: Yes, of course, as a fallback, so to speak.
- P: You can connect via ISDN. But that no longer works
- K: Do they have an old telephone system moulding around somewhere, or what?
- P: No, it works now, it doesn't exist any more. So this port, this facility still exists, it's still there, but I don't know if you can still connect or otherwise. It's only a museum.
- K: Awesome.

- P: But that was funny. Because we sometimes had customers who connected to it because they had such highly sensitive data. And when they wanted to pass this data to this database, they connected directly via ISDN.
- K: Yes, it's actually not that stupid, depending on the situation. You just don't have many problems at that moment.
- P: Exactly. Then we have another special case. This is the new system now. This is our database. And I said earlier that we currently have many servers that only use one database, so we naturally asked ourselves how we can make this database redundant. And unfortunately it is not yet live, the way we have just built it, but we have two databases, the MySQL MariaDB database, which run in parallel and replicate in real time. That works pretty damn fast, too. That means that when a command is executed in one, it is sent in parallel to the other and vice versa.
- K: Okay.
- P: And before that there is a/ There is a special proxy for this. It can also balance the load and is connected before this MySQL. That means, for example, you have/ Strato does something similar. You have a MySQL address, which is now simply the same for everyone and depending on how you connect with the username abbreviation or whatever, it could theoretically take one database network or the other. Or if you only have one, like we do, then you connect to one database. If that fails or is overloaded, he connects the other database. And because we are total backup people/ we have backups more/ definitely many backups. There is a third server that only reads, which means only reads, that only writes, that also backs up the database in real time.
- K: Which is/ Actually only for replication/
- P: It is not connectable from the outside or anything. The is/
- K: He actually makes a permanent live copy, so to speak.
- P: And it hardly has any CPU or anything.
- K: And how does replication work, if you land once on one server and once on another and write data, they replicate each other or what?
- P: No, it's just absolutely in real time. And it's like this. It only works with InnoDB for now. This is because InnoDB already writes a protocol anyway. So everything you write somewhere, a command in the database, read or write, ends up in a log file. And nothing happens other than that this command is simply executed in both cases.
- K: Yes, but you deliberately go once to one and once to the other in terms of load, don't you? What if one of them had, I don't know, an overload?

- P: You mean that's the typical problem that online games like to have? That you can duplicate things or something?
- K: No, that's not what I mean at all. But the moment you say quasi, when quasi any database statement or anything is written, it is written on both at the same time.
- P: Exactly.
- K: But that also means that you can't, if the one, so to speak/
- P: You mean if there's latency?
- K: It is not load balancing in that sense.
- P: But before that there is a load/ so this proxy does this load balancing. Of course/ How can I show that?
- K: But load-balancing in terms of the database means that I write it once in one and once in the other. But you just said that it is always written in both in parallel.
- P: No, load balancing means to me, or is that perhaps just a wording issue, that there is a system in front of it that recognises what system is currently busy in the back and the system that is least busy gets this call.
- K: Yes, exactly.
- P: But that/
- K: But that is still the same database in the end? Logical, isn't it?
- P: No, these are individual systems. But the thing is, when you write the command there, it is also executed directly on the other instance. This means that if you connect to the other one a millisecond later and want to read it, the command has already been executed. But if you now have a write command that would take longer because it needs CPU, then in that case the table is locked, or can only write one page. And unfortunately I have to pass on that a bit, because I don't know exactly how it works. I only know that it works in any case.
- K: Maybe it can then take over the result without doing the same task again or something like that.
- P: Well, that's really all in real time. So that's extremely fast again.
- K: Yes, of course, but just when it/ Then you also have the/ That's why I just wonder just when both or when the/
- P: But if you have a/ If you have a/ So I understand what you mean. When you have a connection and you execute several commands in that connection, that's what you mean, right?

- K: No, no. Just completely, completely normal. I arrive and want to write away a large data set, which would then put the database at full capacity for the moment. And because everyone is doing it at the same time, all the others would then also be used to capacity, so to speak.
- P: Oh, you mean. Now I understand what you mean. Yes, in principle you are right. Yes. I would. But the thing is, the thing doesn't load the database in case of doubt. The thing is that if you have a case where only writing to two computers at the same time is equally busy, then something is wrong with the hardware. Then you are faced with a completely different limit. So/
- K: Oh, that's right, you don't need that when reading out. That's right. Read processes that take a long time don't change any data. Yes, logical, now I have it. Okay, okay.
- P: But why/ So the core of why we do it is actually not to spread the load, but to minimise the risk for failure.
- K: To have the replication, so to speak, or the resilience more or less.
- P: And you can already scale, so. But the case you are referring to would not be website hosting. Then you would have, for example, an application that has two connected databases. One database is only for reading and the other only for writing, for example. So you can have many clients that only take the database or take a node.
- K: Yes, that's right. If you have another case, then you have a kind of Galera Cluster. And then there is also an instance of its own.
- P: Exactly, Galera Load Balancer is that then.
- K: Which is only responsible for replication. Because then you have to iteratively ensure that at the end of the day all bookings are the same in all systems.
- P: And that brings us to the point I was referring to earlier. Many online games have the problem that if you load the server to capacity or the API, then you can fire so many database queries into it that you can duplicate things.
- K: Oh, so then you can sort of move it before the person who is using it or something has gone through.
- P: If in doubt, you can trigger a command that then calculates something plus.
- K: Ah. Yeah, okay.
- P: Almost every online game has this problem.
- K: There we are again in an interesting field. Yes, yes.
- P: OK, that's how the database is set up. That means that the database ultimately runs into Zabbix. And we have set triggers like this. If you look at the same

thing now, you see the network. Then you have the incoming traffic at the firewall. We actually also have this with the [HOSTING PROVIDER] Cloud. [A VM does the firewall and then at [HOSTING PROVIDER] you have the option of doing an internal network and the individual servers are then in the internal network of this [HOSTING PROVIDER] construct.

K: Ah ok. You are creating a virtual network within the [HOSTING PROVIDER] Cloud.

P: Yes.

K: Yeah, cool.

P: And/

K: That is, resource management at the end of the day is really just monitoring and, if needed, more RAM, more storage and more CPUs?

P: Exactly.

K: Yeah, freaky.

P: And now, for example, you have a case where a website, a customer, has a lot of traffic. And this is solved in such a way that if a website has a lot of traffic on this server, first of all the Zabbix server/ Here we come to an interesting and very large area, namely the art of configuring the server in such a way that it also uses its resources or does not overload them. And I can really put my hand in the fire, no matter how heavily our server is used, even if the query then runs into a timing, you will never somehow come to a limit that the entire server stands with us. Because the NginX or the database or whatever is exactly tuned to the hardware.

K: Okay. So if something is led, it is also carried out to the end, in good German, even if it takes longer.

P: Exactly. Yes. So if you now have a limit of 60 seconds of PHP time, then of course it will run into that limit.

K: Yeah well, okay, sure.

P: But if you theoretically have a limit of one hour, then it will also run through and it would not affect the other hostings on this system.

K: Crass. So it is ensured for the time then that the resource is available in any case?

P: Exactly. Yes.

K: Cool. Okay.

- P: I would call it micromanagement or something. And we have really spent years on this. We have written a small script that returns values for an NGINX config that can spit out an NGINX config based on CPU, RAM and a few values that we store there.
- K: So it fits the hardware.
- P: Yes, there are also open source solutions, but we built it ourselves.
- K: That is, there are things in there like how many PHP/ So what are the parameters? PHP threads, uhh/
- P: You have a web, which is a closed system, so to speak. This means that this one website can run PHP so many times at the same time, execute so many NGINX processes at the same time and use so much memory and then, of course, use hard disk memory. And the whole thing also relates to the database, so that a user can only use a certain amount of RAM. So the database is also configured in such a way that the entire RAM is used up to a certain level, i.e. up to 80% or so, in order to use as much RAM as possible.
- K: Oh well, that then ensures that when in doubt from the ENGINX you say, I don't know, five zero something, we're just overloaded before you actually/
- P: That's what I was getting at. That is, this web is a user in the system and we have configured Zabbix, our Zabbix, in such a way that we can see the individual user, how much resources he is currently using. So we don't just break it down to the web server, but to the individual, to the individual website, even the monitoring.
- K: Ah, nice. Yes, yes. Yes, well, you actually want to know, because if someone overuses or something, they also have to know who it is.
- P: Yes, exactly. And that means in that case, in that case we would actually only get an alarm, because we personally/ We could actually almost care less if a website simply uses its entire quota to capacity, then it is actually at least in our/ From our point of view, the problem should first be for a hoster, uhh that's not true from the hoster's side, for a customer to say okay, my web space is now at capacity, I would have to book a higher package something like that.
- K: But I actually notice that via server-side error messages, don't I?
- P: You notice that now, for example/ Yes, so if you have managed to bring the system so far, that's already a lot, then you definitely (not?) have so many customers and so many calls that people will probably complain.
- K: Yes, but these are then quasi/ That is then/ Are they then simply things that run into the timeout, or do you then already get/
- P: [unintelligible], that is/

- K: Does the web server then already say, Attention I am overloaded at the moment?
- P: At some point the web server says, I'm overloaded right now. Try again.
- K: Ah, perfect. So it's not like you just don't get an answer anymore.
- P: That is exactly what I have meant all along.
- K: Because the hardware always matches the config to the real hardware.
- P: Yes. We have only one eye of the needle. That is the bandwidth.
- K: Yeah, okay. If it's too full, sure.
- P: Exactly. When the bandwidth is full, the connection slows down accordingly. We had that last time, because an online shop built such a crazy homepage with three videos or something.
- K: Holy shit. Okay. How big was it then?
- P: I don't know, something like 90 MB or so?
- K: Yes, okay, if you have a lot of traffic there, you'll have a little bit together in no time.
- P: And he really managed to get the bandwidth from this/ So I mean, that's with a [HOSTING PROVIDER] construct/ A [HOSTING PROVIDER]/ The [HOSTING PROVIDER] people don't tell us what kind of bandwidth they really have.
- K: Probably a gigabit like that, right? Round about.
- P: Exactly. So internally they are also connected with fibre optics. But it's just one gigabit, somewhere in the rotation. But theoretically it's also more.
- K: Yes, of course, but that is also captured.
- P: I don't know, no technician can tell me yet.
- K: I think they don't like to say that either, because that's probably also a bit/ Yes/
- P: So. If we make a cut now quite well, yes, you really mustn't tell anyone, but I have this contact with [HOSTING PROVIDER].
- K: Wait a minute, I'll go on/ Pause/ There/
- P: Because I'm just rambling on here all the time.
- K: It was invaluable for my understanding just now. But if you turn around now, you'll put on my shoes. Like this. You want a PHP script that runs in user space at the end of the day. You write a PHP script and with this PHP script you want to try to find out which hoster is the fastest.

- P: Exactly. That's one of the things we've already talked about. So I really do have a few good approaches for it and I've also dealt with it a bit because we've tested various systems. Above all, how the individual processes, the individual ways in which you can utilise a system, how you can measure it or find weak points.
- K: Ah, okay.
- P: And there I have first of all/ First of all, there are four main points, yes, four main points. First of all, CPU and memory, which can be measured, then the file system, the database and the network.
- K: Exactly.
- P: And the individual things can be tested in different ways, namely if you look at it from the CPU point of view, you have different ways of utilising the CPU. If you/
- K: Can you do numerical calculations.
- P: If you do floating computation, numerical computation, then you're using the CPU differently, like if you were basically rendering graphics stuff or something.
- K: Yes, that would also be the question for me. I just want the benchmark that I develop not to be a synthetic benchmark. I don't want to know, so to speak, what the big ones/ You know AnTuTu etc. on the mobile phone somehow. In the meantime, they've gone and got their benchmarks, hold a gaming engine and measure the FPS. Because that's what matters. And so, it should be the goal of my benchmark as well. I don't want to know how many floating point calculations I can now make on the CPU if the average user runs a WordPress where, for example, a floating point operation never takes place or maybe once in a blue moon, you know?
- P: Exactly.
- K: And the question is, do I first have to observe and analyse what a typical WordPress system does? Or could you already tell me from experience?
- P: So I would probably build different scripts. A STANDARD WordPress installation and then a PHP script, which we had already built for example for/
- K: Which triggers typical tasks in there, right?
- P: Which automatically kind of triggers a plus calculation, in the loop from me, 20 seconds and then measures how long it takes, then any string manipulations or string replays or kind of things like that, uh basic loop, just the loop, so without any
- K: i-loops with

- P: and I find if-else stories extremely important. That you randomly generate something and then if-else query something.
- K: Ah, okay.
- P: And/
- K: With random, however, you have to have a high sample size, because otherwise it can run faster with one and not so fast with the other. So random is always dangerous.
- P: Yes, I don't know. You'd have to look at what we did there/ I can't really remember. It was just an if-else thing that actually said, is one or is zero I think or something.
- K: Well, okay.
- P: So the number was random or is bigger. Probably different, is bigger or is smaller or is one or is two or somehow like that or is modulo or things like that. And that would be a script for example. But you can also go here and see what data you get from the system. And there I have experience simply from the CMS Admin Script that I have built. Because I already monitor WordPress sites. And I have a half-finished module that collects data from the system. I have to quickly cheat in what I collect. What there/
- K: I'm going to get something to drink real quick. My mouth is really dry right now.
- P: Me too.
- K: Yes, here I am again.
- P: All right. In fact, this is where it gets interesting. So the thing is, the systems actually always reveal more than you think.
- K: Okay.
- P: I always went here once and simply displayed all the data I find in this CMS Admin Dashboard. From the system, I called it system info.
- K: Yes, I also just wrote it as system information info. Horny.
- P: Yes, but it's now/ You can't see it globally, because for example on our servers, I've disguised the data a bit, for this very reason, but [COMPETITOR 2] and Strato and from there you get this data everywhere. First of all, you get how many CPUs the system has.
- K: Okay.
- P: Then you can find out what the load of the server is.
- K: Really? About PHP?

P: Yes.

K: Crass.

P: And from the first things I programmed, if I recognise that the load is at 90%, I get a notification at the/ I just had 50% once, I got 1 million notifications. Otherwise, I think I'm at about 95%. So/

K: Really? Then it's yes/ That means you can monitor the load on third-party systems?

P: Yes.

K: With every call?

P: Exactly.

K: Wicked.

P: Then you can see at runtime how much RAM is being used. But that doesn't help you so much in this case.

K: Yes, well, there are WordPress plugins that already do that, that always show you the RAM used. I once installed this for a customer because he had brutal problems with his PHP memory limit and so on.

P: Yes, and you can see how big the RAM of the system is. You can see the [unintelligible], i.e. the memory of the system, how the load is. But that's only rough. And then there are a few topics, as you already mentioned, OpCache. OpCache always provides a certain amount of memory per user per resource and a certain number of, uh, objects that can be stored there.

K: So just the limits that are set, right?

P: Exactly. Like the maximum/ So there it's maximum files and the revalidate, um, time or time or something.

K: That's an interval, isn't it?

P: Yes, I/ For our services, for example, two seconds.

K: Okay, but this is some kind of expiry date where he throws the objects away again or what?

P: Exactly.

K: Okay.

P: And for us, the limit is currently 120 MB for the objects. And that is actually quite a lot.

K: Yes, definitely.

- P: And two seconds is also relatively high, um, which unfortunately leads to a few mistakes now and then. But you can deal with that. For example, if you make a WordPress update, then the next call, if your messages don't arrive within, well, two seconds, which can happen, then an error message comes up, because then some/
- K: Oh, because he's still old/ Yeah right.
- P: It's such a classic, you get that quite often.
- K: Do you then also have one, still run a Redis somehow or ne Art/
- P: Um, memcached we have on the, on every system and the object cache we also have on the system, so APCu.
- K: I see.
- P: But I don't think anyone uses Memcached because/
- K: No, by now all I see is always Redis.
- P: Yes, good. So Memcached and Redis/ Redis is really one of those things that you would have to sell to the customers as a package or something, because it's actually its own, closed system, if you want it that way.
- K: Yes, I've actually seen it now with many WordPress specialised hosters that they build it as a fixed component in their stack, because it just takes away so much load in the WordPress environment.
- P: But I am almost certain that Memcached and Redis make little difference.
- K: No, I also don't know why everyone is going to Redis now and not anymore/ Before it was Memcached. Yes. That's right.
- P: Or just APCu. This is the WordPress standard, so to speak, for saving objects. It is anchored in the core. And you have reserved a certain amount of memory. And then you can simply store objects in it with key and data. And you can also get data out of it, which I collect.
- K: I was just going to say that. Can you also read out the limits?
- P: Yes, you can read out limits and then it depends a lot on the config. But as a rule, you can almost always read the server load, i.e. the total thing.
- K: Okay. And of course that's also something that's super, totally exciting. Because you can't fairly compare two systems with each other, where I don't know, one executes everything in real time and the other has no idea, half or the database in RAM, the objects still in RAM and I don't know, maybe part of the frontend code already pre-generated.

- P: Yes, there are a few things I would like to check. And that is, for example, this server-side thing, but I'm nowhere near through the ideas that could be done. You can also connect the MySQL database and the MySQL database also has global data. You usually get the global throughput of the database, how many connections it has and/ or how many connections it has.
- K: Oh, nice. Okay. Yes, something like that, that would of course be for us then to actually survey/
- P: Yes, it is interesting to compare like that.
- K: Exactly. I don't know yet whether I would make it public, so to speak. The data, on the one hand, that we collect the data, so to speak, and on the other, that people then say yes, here, they have this and this and this. But I mean, the moment the hosters realise that it's a differentiating factor, they can disguise it or manipulate it or something.
- P: Yes, let's go back to point 1. The first thing is PHP. This execution of standardised processes is what I call it now. Then the next test is the file system, reading and writing. There you also get data from the hard disk, how the load or throughput is.
- K: Can you read S.M.A.R.T. via PHP?
- P: Probably not. For that you need local there you need/
- K: Do you need real/
- P: Commandline tools. But you can try it out.
- K: Yes, I would have simply written many small files, written a large file with some random numbers and measured that over time. That's how I would have done it, so that you could measure the throughput, the data throughput and the whole thing with reading and writing, of course. But just once because of the access time and once because of the actual writing speed.
- P: Okay. You have, you can do this IO test once. That is, how is the throughput? Then, what takes a lot of time is creating a file, maybe copying it, so copying that into Linux like renaming, copying the file, deleting the file again, and then just I don't know/
- K: A million times. Yes, the problem is that I'm not allowed, as Sebastian from PureHost has already told me, to use these classic benchmarking tools, which are also available, because many hosters seem to prevent them from the outset, because they don't want a new customer to come in somehow and then first of all have to/

- P: Basically, this is what is called a penetration test. You won't find a hoster whose terms and conditions say that you can do a penetration test, because you're always putting the systems at risk.
- K: Yes, that is so. That would be the next question. Let's assume that I now have, as you say, this script that does all these things and you also have a standardised, uniform WordPress installation on top of it, then of course I also want to measure externally, because I also want the peering and the routing etc. to work. That all flows into it as well. But then I actually also have to do load testing.
- P: The question is/ How do you mean/ What do you mean by load testing?
- K: Well, from so let's say 10,000 50,000 100,000 real users are just simulating this website in parallel/.
- P: For example, you are not allowed to do that.
- K: Exactly.
- P: So if you do that, it's tantamount to attacking the hoster.
- K: Exactly, and I know/ I know that some hosters I've already spoken to have said that yes, if we let them know in advance, we can do something like that. So, if I let the hoster know beforehand, some are even open to it. The problem is that, firstly, I want to do these things regularly and, secondly, I don't want to know that the hoster knows that he is being tested. Otherwise, he can of course set the resources to maximum in order to get away with the test.
- P: Yes, but that/ Yes, of course. But that's how it will be if you say, for example/ We have many public projects and there is always a penetration test. And during this period, they always pay special attention to the systems.
- K: Yes, of course, it's normal. I wouldn't do it any other way.
- P: But I don't need/ I wouldn't worry about that to be honest.
- K: Why?
- P: Because the tests may only be so large or have such a scope that they are not actually noticeable in the system.
- K: Exactly.
- P: And now you have to test it a bit. But I'd really like to. I could really support you. Because we've already built something like that in principle. You really have to break it down into file system, database, CPU, memory and network. And what you can do is, what is the average network rate? You can test that.
- K: Yes, what I would do in any case. You can always download the latest WordPress version.

- P: Nah, I would/ The PHP sends a 100 MB file somewhere.
- K: Oh, yes. Right.
- P: Receives a 100MB file and measures the throughput rate. All this doesn't really concern WordPress at the moment.
- K: That's just normal wget, isn't it?
- P: No, I meant this whole thing, all these PHP functions and PHP ideas that I have there.
- K: Yes, the good thing is also if/ The question is also, do I build now quasi for the time being/ But that has nothing to do with the master's thesis now, but is more a strategic question for the feasibility later. Do I build the whole thing on a WordPress basis for now, in order to perhaps have an advantage for the WordPress hosting comparison? Or do I build it purely in PHP from the start, which would also have many advantages?
- P: So I'm going to tell it to you like it is. WordPress is actually totally unimportant in the construct.
- K: Yes, except for a practical test, in inverted commas, when I want to know how quickly a standardised website can be made available. I just want to use a standard WordPress.
- P: Yes, that is again/ That would be a test of it. You could do an installation that/
- K: Exactly.
- P: Ideally, it should be in maintenance, otherwise you'll suddenly have a lot of hacked pages that just run there. So the system must also be password-protected so that no traffic comes from outside or something. Then you can say/ You can easily build a script that measures the runtime of WordPress from start to finish. And then you can think of a few waypoints. How long does it take until WordPress is initiated, until the plugins are loaded, until the theme is loaded.
- K: You can get everything from the hooks, can't you?
- P: Yes, I would think of a few waypoints.
- K: That's a cool idea. Can I reliably determine at which point caching was used?
- P: That is/ No, you can't.
- K: Shit.
- P: The problem is that if you use WPRocket, for example, you have no chance of triggering in there.
- K: Yes, exactly. But the question is, if you don't/
- P: But you can/

- K: call when you send a search query, for example.
- P: You can but/ But caching doesn't really matter again.
- K: But I actually want to bypass caching specifically to make sure that this is a real/
- P: Oh, you mean caching from/
- K: So to make a cachbusting call, so to speak.
- P: I would also rule that out. So a cache busting, I would definitely do that/ I would just put that in. But you meant the hoster, for example, who was it, RaidBoxes, for example, who used the Ngix cache/.
- K: Exactly, they simply have a full-page cache online. And if the page has already been called up once, the second call simply comes directly from the cache.
- P: Yes, exactly.
- K: I mean, I can then try/
- P: If you add a random get parameter, it will probably be enough.
- K: Exactly. I have/
- P: But actually it doesn't matter. You can also write /random something in it.
- K: And there are systems that then ignore certain parameters when they are called. So a good caching system only has, so it doesn't let every parameter bypass its cache, so to speak.
- P: That's why I don't mean a parameter at all, but /random number, which would theoretically lead to a 404.
- K: I see, the path. You're calling it up for the very first time, so to speak, which has never existed before. Yes, you're right.
- P: So there you can build a plugin, for example. What/
- K: Ahso, which then writes in a bit more elaborately on the 404 page.
- P: Wait a minute. No, no. That was for a/ You make some plugin, which is called, uh, test, so /test and everything that comes after that is just wildcard and is irrelevant and is not even considered by the system. That means you always call up the Test page and everything that comes after that is just, is not taken into account by the system at all. You can determine how this is handled with the WordPress rewrite rules.
- K: Yes sure. Yes that/
- P: Then you would definitely have a valid test that always calls a random URL completely. Without get or something.

- K: Right. Then it would be/ Wait a minute. Then it would be a random URL. That is, URL based cannot be cached, then at most it could be the object cache. Yeah, okay.
- P: The object cache is only active when the WordPress is switched on.
- K: Ah, okay. I'll have to see if I can find the/
- P: But you can also go here a little bit and you could also say, look here/ So object cache would only be, you would first have to actively activate it in WordPress with a plugin, the OPcache, which you could theoretically clear with every call. So/
- K: But then you must know something about the system, or can't you?
- P: No. This is a PHP call against the Clear Application.
- K: Oh, cool. Okay, yes, okay, then I would also take that into account in the plug-in, that it is discarded for the time being.
- P: Because this OPcache partially at [COMPETITOR] the OpCache is eight seconds, which is totally insane.
- K: What?
- P: If I then update or copy something over, the page always dies. And that's why I have clear OPcache in the CMS Admins plugin for all triggers that have anything to do with updates or anything else.
- K: Ah. Awesome.
- P: Yes.
- K: You're so far into it. You could probably write this in 1/5 the time I could, this fucking plugin.
- P: Yes.
- K: We'll really have to talk again next year.
- P: I could roughly build that for you. So first I would make an Excel list and see what you want to trigger and I would build different scripts. One tests the database. The other one tests the file system. The other one tests PHP and then you call it one after the other. And at the end you have your test installation of WordPress.
- K: Yes.
- P: But that has to be installed manually somehow. And by the way, you could also do all this with Ansible.
- K: Yes, that would be exciting again. So in the beginning I'll do it manually somehow, of course. The question is whether you'll have to automate it at some

point, because every hoster has a slightly different environment variable and so on. I don't think you can ever completely automate it.

P: Yes, I'd say so. Do you know why? Because you only need the ftp access in principle, namely, or SSH and the database data.

K: Database, username, password. Right.

P: And the path in doubt.

K: Yes, you could even get the path yourself on the first call.

P: And everything else is always the same. Therefore [unintelligible]/

K: Yes, that's right. I just thought I wouldn't go to the trouble of automating it because I actually only do it once per hoster. So my idea was that/

P: Well, but when you have, let's say, 200 hosters, then at some point you want to update your script, and then you update your fingers to the bone.

K: Ahh shit, right. Then of course I want to keep the script and the WordPress version up to date. You're right. Damn. Yes, well, I wanted to use a WordPress remote management software. There are a few out there.

P: Either you do it this way or you make an update script which downloads a ZIP and simply copies it over.

K: Yes. That would work too. Right. ZIP download, ZIP extract, call. Done. Yes, that's right. But we're already deep into the operational side of things. I haven't even got that far with my thinking yet.

P: But, yes exactly, as far as testing is concerned, that would actually already be roughly what I see as the benchmark actually. up to here

K: Yes, but then you know, the problem is that what I still have now is then you know how performant a theoretical single call is. But you won't find out about that now.

P: But I think that's enough to test the system. So to keep an eye on it. And I would really trigger this whole thing every five minutes, every ten minutes or so. Yeah, and then you do an average of four hours. Or or maybe you do a fair 24 hours and the average is then the value videos to evaluate.

K: I would do a bit of data voodoo there anyway. Would, for example. I don't know. I'd throw away the worst concept and so on... Because you've got backups going on at night or something.

P: For example, this will be between 24 and 6:24 at night and 6:00 in the morning, not exactly.

- K: Then I don't want to check anyway and also the very worst values. You have to stay fair somehow.
- P: Whatever. In principle, it would be funny to just throw it into a database and then.
- K: The next thing is the architecture. I want to build the whole thing in such a way, service-based or API-based, that I can theoretically address my script from a central instance of individual hosters, so to speak, then the test runs through and then I get the values back into my database.
- P: But these are now two different approaches, namely the question to me does not really fit what you need. Because if we now monitor our cloud system, then we have real system data and an infinite amount of it. And if you now go here and the closed system from the outside, it comes into the house, the package, as if only what you get to grab and do, properly according to their own standards. That's why it's right. That's what I was talking about earlier. Services are actually not so compatible with that.
- K: No, not for that operationally, but for me, for me. In theory, it's super important.
- P: Important.
- K: Because I have to write here how the systems are set up, that when I then measure, I have to know, so to speak, it is yes, it is yes, actually it is black box testing. I somehow have a PHP script that I can execute. I have access to a database, I can read and write and I can call the thing and run a or or a stack. And the more I know about the black box, the better I can tune the scripts accordingly. I have no idea that I'm not now doing a calculation that theoretically runs for five minutes, because I just know that the average or the normal script runtime is limited anyway, so things like that.
- P: That's why I mean by approaching.
- K: Yes, that's exactly what I would do. I would then just. You can also simply continue via PN and then enter the number of seconds or so. I also want stories like that, that I try longer and longer intervals and see when another success comes back and when does it break off?
- P: I can't tell you that beforehand. You need to know what the script runtime is.
- K: But can you read them out reliably?
- P: There are two runtimes. One is what the web server delivers, so the other X or Y or so. They already have their limit. And then the script. Runtime of the. Beep, beep, beep. Yes, exactly. And you can even read out both of them in principle. You can even read out the ICs. And you should actually be able to read out the LPG script runtime.

- K: Yes, that would be natural.
- P: Maximum values. Yes, because the system already knows all these backup scripts and such. They usually know how the script runtime is and.
- K: They make you so sick. True.
- P: True.
- K: They do it.
- P: Also.
- K: Yes, yes, there are already benchmarking scripts for WordPress or as plugins. Of course, I also look into that. By the way, that's also part of my master's thesis, that I only look at benchmark scripts. Some of them do exactly what you said, just a CPU test, RAM test, if it corresponds to the test blablabla. And then I'll just sort of do that.
- P: You don't have to reinvent that. Probably ne.
- K: But I will analyse it, do a plausibility check and describe it. That's good, because it's not good, I don't know, it's somehow not relevant to practice or never happens in a normal environment or something.
- P: Yes.
- K: Things like that.
- P: Stop. I would. I honestly want to split it up a bit, that you don't have one script, but several small scripts that are self-sufficient in themselves.
- K: Yes, exactly, exactly. And then tests are completed and a main theme runs through the tests.
- P: And then I would really say in the end. The joke of it is, you can make it if, for example, empathy. Shuts off the tap early, for whatever reason. Then the PPP script keeps running in the background until it dies.
- K: That's right, but you're never one.
- P: Are you the classic, if you have a page somewhere, somehow managed to crash it? You can't access their page any more. Have you ever experienced that? Yes, then it's because the process is still running and the PC says "Nope". You actually have a new process running and then you are simply in the world plug.
- K: Yes, but you should actually have some kind of monitoring instance that recognises and terminates such processes.
- P: Yes, that's exactly why I have the CMS plugin now, it's almost a bit of what you want anyway, because I actually have the complete WordPress installation and actually PHP as well. So I go, catch the errors from here, start a runtime

test from the first call to the last. Runtime test from the first call to the last and MSDN with. And if now relatively high, I think 900 seconds if we monitors. Because a lot of pages Becker plug in plugins. And 900 seconds is also enough to monitor them, at least in the frontend. Because if a timeout happens, then it's an error again. Then I tracked with anyway.

K: Ah yes, that's right. And then it breaks off beforehand. You get the response before the time-out comes.

P: And with me the funny thing is, I don't send everything somehow towards the output as Jason or something, but I just send my Epi Script directly an Epi.

K: Every single one. Or then spellbound at the end.

P: Every single one.

K: Rad. Okay, how many requests are you asking for?

P: That's not so much, because the monitoring is not done with I don't monitor the performance, but I only monitor the errors. And if, for example, the worst thing that can happen is that rabies triggers an error and there are 80 people on it or so. Then I would. By the way, I've already managed to get 30,000 emails within an hour.

K: But generate an email to everyone.

P: That was okay back then, too.

K: That was.

P: That we now have 30,000 database entries and I already catch that, that I remember how many errors are triggered. So yes, okay, I save that. In the meantime, I also save the last 20 errors in the database and display them in WordPress.

K: Local. Yeah, okay, that's cool too, of course.

P: This is my favourite function. If anything in front of a monitor. Yeah, right. We forgot something. You can still check the mail function.

K: But to what extent is that relevant, whether you can use it or what?

P: Yes.

K: Yes, I wanted to leave the subject of email hosting out of it for now, because actually, if you're honest, if you have a decent, well, if you have a client who's not a total idiot and it's a website, then they need to be anyway.

P: GoDB.

K: Compliant email archiving, etc.

- P: And I don't mean email hosting. Sure, you could test that too. I meant because it works. Sending a mail to the server.
- K: But I would also really leave it out, because I think most of them then maintain installed TB and that's good.
- P: They are also a bit safe.
- K: I would even say that if you can send mails from WordPress via Sendmail, then it's actually rather a bad sign.
- P: Or? So, as a general rule, nothing. Yes, but it's good with me.
- K: There's nothing wrong with it if it's properly configured and all, but it's a potential vulnerability.
- P: If I am not relevant for this test for my customers, because I have the case relatively often that even with the TP plugin or so a mail is not sent. Ah okay. And I catch the error. If it is programmed with STANDARD, I catch the error and the mail, then with me in the API. Yeah, so.
- K: On the other hand, as a hoster, I would be interested in not allowing sendmail, because as soon as something is hacked, you're immediately on the skid.
- P: We have solved this a little differently. We have a maximum throughput per user per hour in the server. That also prevents someone from sending newsletters or something. Yes, yes.

A.2 Interviewee 2: Max

Interviewer: Kai Spriestersbach (K)

Interviewee: Managing Director (M)

Date and time: December 16th 2022 14:48

Location: Virtual meeting via Google Meet

- M: Okay. Recording in progress.
- K: Good. So, exactly / The first question would be, so to speak, so the request to you to take me once very briefly, so what your stack looks like, really from the hardware that you have somewhere on the shelf, up to all levels, operating system, virtualisation, customer management, yes, up to the web server, so to speak.
- M: Okay. I'm only talking about our own hardware now, because we are aware that what we still have on board with a third-party provider, namely [HOSTING PROVIDER], we will gradually sort out next year, because simply our own

availability has been higher for four years than what [HOSTING PROVIDER] offers us.

K: Okay. Nice.

M: Yeah. So we have on our own hardware an availability of well over 99.99% on an annual average. And/

K: You do housing somewhere in the data centre?

M: Exactly. So I explain it, I can also share my screen with you. I can explain it better there.

K: Yes, with pleasure. With pleasure.

M: Good. Okay, then I'll share that one. I think that will be recorded as well. So, chop, chop. So, what do we do? At the moment we are running servers in actually four different data centres, two of which I don't want to go into too much right now. Those are the two from [HOSTING PROVIDER] in Nuremberg and Falkenstein.

K: Yes, exactly. I know them too. Yes.

M: Exactly. We'll leave them out of it. What we've been doing since 2018 is setting up our own hardware in a data centre here in our [STATE], which is the data centre of a local network operator. In the meantime, we have a good rack full of hardware there, and since the end of last year, we have been doing exactly the same thing from a technical point of view at a second location in [STATE], namely at the [NAME IT 1] data centre. This is a very, very new computer centre. I think it's about four years old now. There's also an image film and all kinds of things here, and the hardware is provided by a local IT systems company just around the corner from us, namely [NAME IT 2]. They've been around for 30 years. They have 160 employees and do nothing but Windows servers, virtualisation, etc. and have also placed the VMware-based hardware there. That means, to explain the stack, we have, um, VMware is our virtualisation layer for the host systems, i.e. for the computing units, and then we have a software-defined storage, a centralised storage. This is virtualised with Data Core. And that means we have an HA solution running. So a whole host can fail, on each side. Several disks can fail, CPUs can fail, everything can fail on each side, sometimes several times, and the operation is not affected. The whole thing is so blatant in day-to-day business that if a complete host system, where we sometimes really have 20 to 30 VMs on it, i.e. virtual servers, fails, the complete VM is started on another host within less than a minute, because the database is the central storage.

K: Yes.

M: So VMware just says, oh crap, the host here just failed? Hey, host number two please take over the VM from the central storage to the other host and then they will go up again. That's happened so far I think twice since we got the hardware there and that was really/my downtime monitoring all went down and within a minute they were all up again. That's brilliant, that's really, really cool. And that's what [HOSTING PROVIDER] doesn't offer now. So [HOSTING PROVIDER], if something goes down, then they have to migrate. We had a downtime this week, 17 minutes, and they wrote that they were being migrated to another host. That took 17 minutes for a VM.

K : Yes, of course. Yes, well they offer it if you are in the cloud or if you run your own Proxmox on it. Then you can also do it with [HOSTING PROVIDER] servers. But of course you need the virtualisation solution. So that's exciting. That means that in your VM is quasi CPU and RAM and probably also the database or? And the memory is actually the file storage or what? How is that there?

M: It's, no, it's everything. Uh, hold on a second. I can hook you up real quick. I can take here in the Google thing, I can draw meanwhile, it's actually everyone, you can do everything.

K: There is now such a/

M: Like this. So. (...) We have, well sucks, I'm not familiar with the tool here. We have up here we have two hosts. So when I talk about a host I'm really talking about a physical server. So. VMware 1, VMware 2. Do you know a bit about VMware?

K: Um. Theoretically yes. Practically, no.

M: Hmm, VMware is just the virtualisation software in the Windows industry. Whereas meanwhile Microsoft/

K: That means Windows servers are actually with you?

M: Huh, no. No. No.

K: No? Okay.

M: So, I have to/ We have two host systems here. They are connected in a network and the VMware VCenter is located there on the local level. This is practically their administration interface. And here in this VMware software we can create a new virtual machine with one click and install any operating system. This can be Windows. But in our case it's Linux.

K: Okay.

M: And then VMware wants to know, hey, okay, you want to create a new VM now. How much CPU should it have, give me x cores, how much RAM. Ok,

this much RAM. So each of our hosts always has 24 cores physically, two CPUs with 24 cores each and the computing unit is practically up here and the storage is down here. And then VMware asks, OK, where is your hard disk? And then we practically have a virtually connected hard disk from down there, these two servers, which also practically mirror each other in a network. Each of these servers has at least four hard disks in it, four NVME hard disks, which in turn make a raid and then an overlapping raid over both host systems.

K: Okay, but they are connected via Gigabit Ethernet, so to speak?

M: Right, I haven't drawn that in here now, but/

K: Yeah no, I'm just asking.

M: For each host you have two ten Gigabit slots and they are practically all connected crossover with each other, so that you can always compensate if a card fails, an interface fails and so on, right. Yes, and the great thing about our construct is that it is growing in width. We now have eleven of these VMware hosts in one data centre and four in the other. I just ordered another four yesterday. And since everything is managed centrally via this vCenter, several hosts can fail later and it will still distribute all the VMs to the other hosts.

K: Yes, exactly, I know that.

M: Yes, exactly. It is similar with the Datacore. The Datacore lives through the number of hard disks. The more hard disks there are, the better the performance distribution, the higher the reliability. And now we come to the only Windows component, the Datacore control software. It's practically in the Windows machine again. I wouldn't know if the function of the datacores would be impaired if the Windows machines were to crash briefly. The datacores have never failed in four years, so I can't tell you anything about that.

K: Okay.

M: Good luck. But as I said, it could be here now, we've had that before, it could be a host failing here now, it's not a problem because they're practically mirroring each other all the time. This is really HA at its best. So much more is not possible. And it's also really well performing, because that was always the problem when we talked about Ceph storage. I'm sure you're familiar with it. C. E. P. H., Ceph Storage.

K: Oh, Ceph, yes exactly.

M: A lot of people use it. But it was never as well performing as what we experienced with Datacore, because Datacore is a software defined storage.

K: Yes, the access figures are much better than the/

M: Yes, so the/

K: latency.

M: Exactly right. That was just/

K: Okay.

M: Because of the software, the data is read and written away very differently. And yes, we did a lot of tests and analyses at that time before we really put customers into it productively. And I really have to say that since we've had it in the form we have it now, which is now almost four years, it's been so incredibly stable that we've now decided that we'll gradually move almost all [HOSTING PROVIDER] customers to us starting next year.

K: Ah yes, cool.

M: So we are talking now / At [HOSTING PROVIDER] alone we have 160 servers running in the cloud by now. And we're going to move those into our own data centre as well, because the hardware here in our computers, that's ours, that's really, we buy that. The only thing we pay in rent, practically on a permanent basis, is of course the electricity connection and the licence fees for VMware, you know, for support, suspension and so on. But the hardware, that really belongs to us.

K: [unintelligible] Okay, nice.

M: Excuse me?

K: You actually buy it and write it off. So that's kind of yours then?

M: Right.

K: Wow, okay.

M: And now, here's the big kicker. Everybody's raising prices and we're going to go next year and announce a price cut for our servers.

K: Oh, rad. Okay.

M: Because by going to our own hardware now and not renting somewhere externally, even though [HOSTING PROVIDER] is already very cheap, but still at [HOSTING PROVIDER], we pay every month and actually have nothing. If we would stop giving them money from one day to the next, they would say okay, you don't have anything anymore.

K: Hmm, yes.

M: And through this strategy and through a certain adjustment of the VM sizes as well / As an example, at our costs, the smallest server has 4 cores, 16 GB RAM and 160 gigabytes of storage space. That's / The RAM and the hard drive almost nobody needs. People want to have the cores, but they would also manage with eight gigabytes of RAM and they would also manage with 50 gigabytes

of storage space. And because of that, we practically adjust our tariffs and say okay, less than before in terms of performance, but also the price.

K: Ahh, yes, yes, okay. And that is probably still more than sufficient for most people.

M: Right. And anyone can also upgrade individual components via our interface in the customer centre. They can upgrade the CPUs, upgrade the RAM and the hard disks.

K: That's very good. Okay, okay.

M: Even now, that is already possible. We have a tariff, we call it Configure Cloudserver. This is the Flextower, it has a basic configuration, four cores, 8 GB RAM, 50 GB storage, exactly what I just told you. You can then say, no, watch this, I need six cores, I need 32 GB of RAM and I need, I don't know, 200 GB of memory, and here you have your price, no.

K: Okay, now we come exactly to the exciting questions. Great.

M: Okay. Yes. Then shoot/

K: So, there would be now/ Let's first of all the topic really managed server and then later again the topic in inverted commas hosting, because there it runs a bit differently. Okay, that means here, if I now say, I would like to have, I would like to have four cores. That means that I get these four cores reserved exclusively by your host, so to speak, which are only available to me, or are they dynamic cores?

M: They are dynamic cores, whereby the host is monitored in such a way that it is never more than 60% utilised. So we set an alarm through our monitoring/ We are practically alerted when we notice that the host is x percent utilised, we are alerted and, if necessary, we would also move the virtual machine directly in live operation without downtime. In other words, we do not give dedicated cores. What is dedicated for us is the RAM, which has to be dedicated for VMware in our case. And of course, I think the hard disk is also dedicated, which we provide, but the cores are not. But we give it to them and then practically make sure via monitoring that a host system can never be used to, I don't know, 80, 90% capacity, because then at some point it will affect all customers.

K: Yes, exactly. That's exactly it. Okay. Yes, good. Otherwise you wouldn't be able to offer the prices, because real reserved cores would mean that after six customers with four cores, the 24 core would be reserved.

M: Right.

K: processor would be closed and then you would need a new box. So/

- M: Right. And first of all, that would be a waste of resources, well, from an ecological point of view.
- K: Yes, they are bored most of the time. Exactly.
- M: Yes, that's how it is and it would also be, yes, then you would really have to offer prices for/ We had that in the past. We started earlier with dedicated cores for the servers, but then we realised, hey, it's not a difference in performance. So it's not that it's faster, because if you have a shared core, you monitor it cleanly in the background for utilisation, you end up with the same thing.
- K: Yes, of course, one is always free. Then you get it immediately. So from there.
- M: Exactly right. The CPU works on the clockwork principle. Do you know your way around?
- K: Yes, exactly.
- M: That practically the requests are always, always running in circles and the more people are practically at the individual clock times, the longer it takes. And that's why we have a strategy that we invest massively in the hardware to make sure that each host, my wish would be, is not more than 50% busy. So that's my wish, that we get there. And then, through this over-provisioning, say, okay, but we also take on more customers than there are physical cores available. But because we have so many host systems, no customer is affected. And the great thing is that he also has really fat, expensive CPUs/ We have clock speeds in the normal clock, uhh, clock speeds of 3.0 gigahertz per core and with hyperthreading we have 4GHz. And if you go to any cloud providers so example [HOSTING PROVIDER], they clock in the range of 2 to 2.4 GHz.
- K: Yeah, we talked about that before, that it actually makes a real difference for single-threaded PHP. You had said that before. I think that's also a very, very exciting topic. I'll write a little bit about it. But that's not really relevant for me right now. The exciting question now is that, on the one hand, virtualisation and virtual machines basically ensure that enough cores are available, even though they are not dedicated. And that means that at the moment when a VM is too much/no, you don't have to migrate it at all. There are always enough resources. But that means they are within this one VM on this one host, um, is its own pool, so to speak. Or it's the/ Are all the hosts one big pool?
- M: Um. Yes. Yep. So through this vCenter interface, you have all the hosts listed. And you can also really say, show me now the performance across all hosts. So you practically have a CPU graph and it shows you all the available gigahertz numbers, it then shows you how much you are currently using across all the hosts. You also have exactly the same view per host system and per VM. So you have practically in several levels, you can look at that.

- K: Okay, aggregated upwards, so to speak, yes. But that means the system doesn't actively move VMs to other hosts, you have to do that manually?
- M: Exactly, we do that manually. I think there is an additional module from VMware, which costs a lot of money, so that they do it automatically. But since our guys have built up such a great monitoring system, we get 24 hours a day, 365 days a year, and we also have a defined standby around the clock. We always notice when a host reaches a limit. And I have to say honestly, I don't know when the last time was that a host reached its limit. Honestly, it is/
- K: Yes?
- M: It's so neatly distributed. So we really have a person who has to make sure that whenever a new VM is created in the system, where it is placed, on which host, so that all hosts are equally utilised on average, right? I just wanted to see if I could get onto our VMware system really quickly. But it's not that easy, because everything is only accessible via certain VPN tunnels.
- K: No, no. That's not so important. For me it's about the question, the coordination, the VM, the single one, on which the web server of a customer runs, so to speak. It has to be more or less over/ You are, I think, on NGINX only, right?
- M: No, we do the hybrid mode. NGINX is the reverse proxy that sits at the front.
- K: Ah yes, okay and Apache/
- M: All right, that's where I am. All static files except the php processing and the Htaccess file is also read by Apache. Yes, that's just the comfortable solution for the customers, because almost everybody works with htaccess all the time.
- K: Yes, okay. I was about to say, it's probably just because of the htaccess, yeah okay. Yeah no, sure.
- M: Exactly. PHP is more practical stand-alone PHP-FPM anyway. It's practically its own stand-alone module. That means, whether you control it via NGINX or Apache, it makes no difference in terms of performance. It's its own socket, or whatever they call it. I don't know, so/
- K: Exactly, it's connected directly, okay. But that means that it's also your art, so to speak, to make the config like this/ So to what extent do you then restrict something like PHP threads or something, so that in case of doubt you get an error message, five zero something, and don't run into timeouts? Because that's actually the danger that the hardware no longer responds before the web server can actually say, "Wait a minute, I'm under too much stress here.
- M: Exactly. So we do that/ We have to differentiate. We have the tariff line, one customer, one server and we have the tariff line, several customers, one server. But it has to be said that even if you have the tariff line, one customer, one server

and, for example, it's an agency and they have 50 websites, it's technically the same. We isolate each individual WordPress website from the others. We do this with a virtualisation layer from CloudLinux, it's called, and in this way we ensure that one website can never affect all the others, i.e. that the entire VM can be used to capacity. We limit CPU, RAM, hard disks, I/O and we can even limit the number of incoming requests that practically bypass the cache. That means, for example, if you have a template, nice example News Paper Theme, you probably know.

K: [unintelligible] No. Doesn't mean anything to me now.

M: Yes, it's a very, very common template for news websites and Google WordPress Newspaper. And they have a viewcounter built in by default. Such an AJAX...

K: Ah, yes.

M: Exactly. I just say Admin Ajax. And then we have clients who have some article that goes through the roof, featured in Google News or whatever. And then all requests on their website basically go through our cache. But then Admin Ajax always comes and reloads this shitty view counter. And then we really have had the effect that this view counter opens so many PHP requests to the server that in the end the whole customer has shut itself down. And we can limit that by saying okay, you are only allowed to start so many PHP requests per minute.

K: Contains the limit, so to speak, the number of PHP threads, but the connections, the actual TCPIP connections don't, do they?

M: No, not those. We don't limit those at all.

K: Okay. Okay. And the moment no more threads are available, the server then throws a 500, right?

M: Good question which 500 that is.

K: But any one probably, probably [unintelligible]

M: 502 or 503 would have to be.

K: Yeah, okay.

M: It also says that the resource/ Ahja, Resource Limit Reached. Wait, I have to see which one that is. This is the 508 Resource Limit Reached, that's what it's called. Exactly. Then the server at the back is still fully performant for all other customers, but this one customer has practically taken hops for himself in what we call his LVE, Lightweight Virtual Environment.

K: Yes, okay. At that moment, something probably goes on in your monitoring and that is probably a moment when you either help the customer so that it needs fewer resources or probably upgrade him, right?

M: Uh, at server level yes. But that seldom happens, because the servers are certainly seldom used to capacity. On the website level, the monitoring will only strike us if our monitoring no longer receives a response from the website. That means that we really have to monitor the homepage with a normal monitor like Pingdom etc.. monitor. And then the start page should really no longer work so that our monitoring is triggered and then our guys also notice this and see what's going on. But if only the website is slow and instead of one second it suddenly loads 8, 9, 10 seconds, we don't necessarily notice that, because the customer could try something himself. So that with 5,000 web pages, the monitors hit so often/

K: You can't sleep any more. Yes.

M: That's too crass, that's too crass.

K: So that means that just these eight, nine seconds can come about at all with you? Don't they all run into a time-out somewhere beforehand?

M: I don't know how many seconds the time-out is defined, but definitely over ten seconds is a time-out for sure.

K: Okay, okay, nice.

M: Yeah. Because you really have customers, especially agencies, who book the server with us and move their own customers themselves without our service. They forget to install a caching system for their customers and then the WordPress is so full that they have 10 or 15 seconds response time "out of the box".

K: Okay.

M: We have a monitoring system for this, our own programmed one, so we scan all the Plesk domains that are on all the servers with us, we scan them several times a day and log the response time and practically do an average analysis of the response times of all the websites with us on the system. We see that I can now/ So at a glance I can tell here in our system which of our websites are currently slow and that's where the next step is that we proactively approach the customers from January onwards. The slowest first and practically look at them, say hey, we've noticed that your website is very slow. Should we see if we can activate our RocketCache? Because often there might be reasons why the agency says, no, no, please don't install a plugin, because we have integrated something dynamic and that is always very individual.

K: Yes, classics, I know them too, unfortunately.

M: Exactly. But as a premium provider, we want to say that we proactively approach them. I don't think I've seen that anywhere before, that a provider

monitors the performance and then actively approaches the customer and says, hey, your website is slow. We could make it faster.

K: Yes, okay. So you don't just put forced caching on the server by default, like other providers do? If the customer doesn't want that, then it doesn't exist.

M: Right. So let me put it this way, the classic new customer with us naturally books our relocation service for his website. This means that one of our employees will move the website from the old hoster to us using a very, very granular checklist. And of course he also installs WP Rocket afterwards, because we also buy the licences, and configures it the way we think it should be configured. And we also send the customer, every customer, a before and after GTMetrix comparison as part of the relocation service. So every customer practically gets a visual handout.

K: Test you then actually the entire functionality? Because then you really have to click through all the forms, i.e. everything that is dynamic, at least.

M: Yes, well, it is/ I know what you want to say and it is also very, very complex to analyse everything. A script can really not work because of this deferring and so on. We always do visual tests on a few subpages, the start page, a few subpages and always write to the client, hey, we have now optimised the caching etc. Please check the functionality of your forms. Please check the functionality of your forms. We do that anyway. Okay, the mail server also has to be released/

K: Yes, of course. The things have to arrive first. Everything. Yes.

M: Exactly. That's why we always write in, please test the functionality of your forms, whether the e-mails arrive and so on. Of course, we always leave that a bit in the hands of the customer, because with a 25 € hosting we can't do what we already do there with the relocation service, the performance optimisation/

K: Yes of course.

M: We can't test every form and then run after the customers to see if the test email has arrived. That would not be feasible.

K: Yes. That's why I'm asking, because especially with custom solutions or when something is built with form builders, you don't know how it should behave, especially with error messages and so on. So you can't measure it at all because you don't know what. No, ok. But then of course, point it out to the customer. Okay, great.

M: So we do visual testing in any case and my guys also all have the order that functionality comes before performance, that means just to guarantee 100% PageSpeed, it can't be that the website doesn't build up cleanly anymore, or something. Sure.

- K: Okay. That means for me, now I want to turn the tables, so to speak. I now have access to a host system with you. It doesn't matter whether it's a vServer or a server for one customer or a server for several customers. And now I would say, I'll start a PHP script to measure what's going on on your server. How could I use this script, um, so let's say you suppress known benchmarking tools that you just don't want running on your server?
- M: No.
- K: But the customers, do they have/ Exactly wait, I have to take a step back. Are there clients that have SSH access?
- M: Um, only the, so the server customers have CHRootet SSH access. That's practically a neutered SSH and whether the normal customers have it, I'm not sure at the moment, but you can't do much about it. But I would have to look in our help article to find out what exactly that is.
- K: No, that's enough for me. Because my approach will be anyway, I said at some point, I want to develop a benchmark that should run on every badly castrated cheap hosting, that is, it can only run PHP anyway and there would be/ Sure, you can use any other command line tools and benchmarking, there are enough CPU benchmarks etc. But that doesn't make sense either. But that doesn't make sense either. And my goal will also be, let's say, to cover the 90% case, i.e. to do realistic things. I don't want to calculate something that a customer will never need.
- M: Yes. Exactly.
- K: Or, I don't know, any, constantly writing and reading five gigabyte files, which almost never happens. That's nonsense.
- M: Exactly. So that's what I always say when people come with their, there's a WordPress plugin/ I don't know what it's called.
- K: Exactly. The WP Performance, I think or/
- M: There are two. One is the WP Performance Tester and there is something new now. It does a bit more graphical visualisation, but it mainly tests the hard disk a bit, etc. And that's not very realistic. And that's not very realistic, because what the normal user notices in WordPress, I'll put it this way, the visitor notices cache at best, like this. That's fast. But now for example you, you have editors probably jumping around in your system, you notice the backend.
- K: Right.
- M: And in order to have a realistic test, you actually just have to measure the response time of the backend regularly. And I do that for example via the Admin Ajax, because via the Admin Ajax you basically see how the WordPress

responds. Of course, what goes into the Admin Ajax is how bloated a WordPress is.

- K: Exactly, but we can do that relatively well. The problem is just that, for our real comparison, it's not a problem because then I install the same WordPress on all hosts, so to speak. It is standardised, then it is also comparable. The only problem is that I also want to publish a plugin so that people can install it on their own systems and make comparisons of how my current system compares to XY. And then, of course, I can't assume that their WordPress is so different. That's why I've now thought about building a small PHP script dedicated to this case, which simply does a few standardised tests. This is a slimmed-down version, so to speak. Okay, but that means that for you that would be/ You don't have things like soft limits or hard limits that you say, I don't know, if there's a WordPress update where the entire core is replaced, then a script can run for I don't know, three minutes before the time out comes?
- M: Ten minutes is our standard time, 600 seconds. And if a customer needs more, he should write to us in support. But ten minutes is standard for us.
- K: For script runtime?
- M: Yes.
- K: Awesome, that's insanely long. Okay, but good. Yeah, sure. But you limit it now via the threads, so to speak?
- M: That/ So the background why we made the ten minutes is simply because all these premium templates with their installers take so long. And if you set it down to, I don't know, 60 seconds, 90 seconds, you have so many support tickets because of shit like that.
- K: Yeah, all the stupid backup plugins that nobody needs, they don't work either.
- M: And that's also the first/ But that's also the first impression that the customer gets. He comes to us on the platform, wants to install a template and then he notices straight away that it always breaks. That is frustration for the customer. And so we said, hey, look, what does it hurt us if the script runs for 10 minutes? He's in an LVE anyway, he can't knock us down completely anyway.
- K: Yeah, that's true. The problem is only a bit in inverted commas with things that get out of hand, that the threads could be filled up relatively quickly, if you trigger something or something runs stupidly slow, then your ten slots or however many you have are half full in no time and hopefully you get that as a customer and then always have to check what's going on. That's always the argument I hear on the other side, those who have a script runtime of 30 seconds, for example, who say that if a script takes longer than 30 seconds, then it's not a normal request from I want to look at something in the frontend, so to

speaking, and we have to guarantee that the threads are made free again as quickly as possible. So it's just a bit of a balancing act.

M: Yes, so you really have to test your script with us afterwards. So I can't say that now.

K: It is the question how I interpret it. So that's the next but one step that I do in the derivation. But/

M: I prefer to do it externally. I take the realistic conditions that a visitor or a backend user feels. I do it through the admin ajax file. For example, I have, I'm not really allowed to say out loud, but you know [BLOGGER], ne?

K: Mhm.

M: He did a WordPress comparison, the hosting comparison, and the [COMPETITOR 1] is at the top, first place, the best and here and there and so on. As it happens, the [BLOGGER] is now also hosted by him and you know, the commission. It's all good. But he really has a [BLOGGER] there. Did what you want to do. He has pulled up one and the same WordPress installation with us, with [COMPETITOR 1], [COMPETITOR 3] and [COMPETITOR 2]. But he didn't publish the URL anywhere. I then found out. For all hosts I found out and then I said with my monitoring okay, I measure the response time of this backend via Admin Ajax. Because what's the point if I measure the response time of a cached frontend? It's so low and always in shape. I measure the backend/

K: [unintelligible] optimised in the end. They actually only optimise for cached requests with the TTFP. That's all they do.

M: Exactly. So. And now I have realistic average values and then I see exactly, if there is a downtime I get an email, so I know, ah the hoster here has had a downtime.

K: Are they permanently online or has he taken them down again?

M: They are permanently online.

K: I see. He runs them permanently.

M: He runs them permanently, does long-term tests exactly. He does it with Uptime Robot. That's what he says on his website. But I just / He just wrote, yes, [COMPETITOR 1] was the fastest provider or something.

K: Yes well, you can do that/

M: Then I stopped/

K: He can claim that if the day is long.

- M: Because I also know how [COMPETITOR 1] acts in the background, they are cooking on a very low flame, right. Like yes/ They rarely write 2800 customers and if you do a bit of reverse IP/reverse DNS, you get 500 domains running there, domains, right. And in the same way, they have distributed everything they have on four [HOSTING PROVIDER] servers. When the day X comes, it could possibly get nasty with the backup. But good. Yes, as I said, I would always measure it out via this Admin Ajax or other comparable things and not with any I/O benchmarks on the server. That's not always meaningful.
- K: Yeah. So at the moment when you have a standardised system, I would definitely do it that way. So for you, we can also talk openly about it, it will definitely be like that, that there is just the standard installation. But I don't want to do it anonymously, so that the hosters don't know about it. Of course, I think it's also in everyone's interest to be fair and not to send the cheapest package all at once to a dedicated machine just for the benchmark.
- M: Yes, all good.
- K: There are definitely hosters that I trust to do it right away. And the deal should be like this: for every package or every tariff that we benchmark with a hoster, the hoster has to write us a monthly credit note for the monthly amount and can then be sure that we will book it again under another company. That it remains cost-neutral in principle, because at some point/ I now thought, with you for example, I would definitely want to benchmark both a dedicated vServer and the shared hosting differently, because otherwise I don't think it makes sense to compare it. You don't have apples with apples and apples with oranges. And then there are different tests. There is the one test, as you say, backend is very important, then frontend cache, frontend uncached, from our servers and of course the whole thing again via external servers. Uh leads us to the last point. How do you deal with it if I want to run a load test, which is also a bit theoretical, you are not allowed to do it, because it could potentially be considered a DDos attack. Of course, it can also be intercepted somewhere by the firewall, whatever. Um, do you think it makes sense, because it would of course be very very important for me, because a site that is, let's say, super super fast, but somehow allows five PHP threads and then shuts down, I have to evaluate differently than a site that potentially, I don't know, manages 10,000 visitors at the same time, but is perhaps minimally slower for individual visitors and in order to stop/ The question would be, what would happen now if I were to start with/ What are their names now? They have changed their name.
- M: Loader I/O?
- K: Yeah exactly, Loader or the others. Hang on, I'll tell you in a minute.
- M: Yeah, I know who you mean. I know who you mean.

- K: I would try to make a cooperation with one of these providers. Um. So where do we have it. The problem is that I don't want to do it with a machine, because then you don't have it realistically. There are also these classic load tests from the command line and so on. That's also nonsense. Load testing tools. Where is it? It used to be called Load Impact. What is it called now? Ah K6. Exactly K6.io.
- M: Okay.
- K: Exactly. They are what I was looking for in my research, so to speak, they were the most sophisticated ones. The question is what happens when I unleash this thing on you guys? Will the alarm bells go off? Would you have to tell me beforehand?
- M: It really depends on how crass the test is. Loader.io doesn't usually set off alarm bells when you test for static elements or something. Of course, if you send 100 requests per minute past the cache, it could be that the website goes down and then the downtime monitoring will start. But as a rule, other customers do it this way, when they really let a pentest company in, they always tell us beforehand so that we can deactivate our monitoring if necessary or put it on pause or something. But I don't think much will happen now with the usual monitoring, I'd say. So as I said, of course, if you go past the cache, the website will stop responding at some point because the LVE says stop, that's enough, that's all I'm doing.
- K: But the information would also be very interesting for me. So I definitely have to consider the scenario/ Of course, the cache is great for hopefully most of the users who come, who go to the Cache. But for me, in order to really balance the limits of the server a bit, the uncached variant is also super important, how much can go in parallel, so to speak.
- M: Mhm. Yes.
- K: I just want to start and I will notice when the servers react accordingly, you have to contact them at the latest.
- M: Yes of course, so I can't tell you much before, you just have to do it. Of course I want you to have a realistic test from us. What's always important to me with these tests is that the test procedure really corresponds to what will happen in practice later, that is, simply spoken, what a rush of visitors to the website would cause, that's what a test has to reflect and not somehow, ah well, the hard drive at [COMPANY] only wrote away 50 megabytes per second, because for example our storage is designed for reading, it's not designed for writing. That means that when you write something on our system, it's not fast, because WordPress is more than 75% a read application. It's always reading and those are just little things, but/

- K: That's something you can/ You can rest assured. I will definitely take you on board again in the next step, when it comes to the concrete development. Because it's similar with the database. I have already experienced that some people, when they cluster, for example, also scale to the main read. And it's just a box that writes. Because most things are really read-only. But that also makes sense and I want to/ So my claim is actually to make a realistic testing that is as little as possible somehow synthetic or artificial. Because everything else is just nonsense.
- M: Yes, I am/ So, if you want feedback from me beforehand, just about the test procedure, regardless of where you test it with us, as I said, I'd be happy to give it to you. I'll be happy to give you that. And otherwise, of course, I also want you to do a realistic test with us.
- K: Then I'll just send you the Master's thesis at the end and then you just give me feedback, because in it I'll sort of describe my approaches.
- M: Yes, okay, gladly.
- K: The master thesis is to look a little bit, how are modern hosting systems built, what is there, in which variations, how is performance monitored, just like what we have talked about, also dynamically, controlled, allocated, managed, also limited, so also resource limitation, so to speak, and what does it mean for a possible benchmark approach.
- M: Yes, very good.
- K: Let's see. I don't know, maybe I'll save myself the work, this whole thing, what would be a realistic visitor and so on. I will probably save that in the context of the thesis, because then it will be exorbitantly more complicated. If you then say, actually, these load tests must not come from x URLs, and so on. But I think that in the Master's thesis I will only work through the script that actually runs on the host. And the external tests, I'll also use the usual tools that exist, the commercial ones. It doesn't make much sense scientifically, and that's another topic. But we can talk about that again. So I will really use something like loader.io. I will also look, yes, I don't like Google PageSpeed Insights anymore.
- M: No, that's not a speed test, but the source code of the website and the scripts are analysed. This has nothing to do with the performance of the hoster, unless the hoster is so slow that even the initial response time from the server sucks.
- K: Right. Exactly. And the main influencing factors, which from my point of view is also correct from a user experience point of view, are now just things like the theme and the JavaScript stories or just generally whether the thing is set up in such a way that it just casts or doesn't cast. But that's not really relevant to hosting in that sense any more.

M: Yes. So that's/ That's what we do with our relocation service to the customer. Of course, our server is usually faster than the old provider. Then we activate caching, which he didn't have before, then of course we are superheroes. But what we also do, of course, is really do basic core web vitals optimisation with WPRocket and that's just really cool. I don't know if you've seen that before with us. This demo move, which we now feature quite strongly, directly on the Demo Move homepage, you really pick up the customers. People come, regardless of the hoster, and you show them, look, this is your current hoster, this is what it would be like with us, both visually with this GTMetrix and they get a link to the cloned environment with us on an isolated server and can really click through and so on.

K: Yes, that is really cool.

M: That goes down really well. It goes down really well. So we've won a lot of new customers who said before, why should I go to [COMPANY]? They are so fucking expensive, there can't be that much difference to my current one. So we say, hey, watch out, just do the free demo move, then you'll see if it's worth it for you. And we have a conversion rate of 50% from those who book the move.

A.3 Interviewee 3: Andy

Interviewer: Kai Priestersbach (K)

Interviewee: Customer Success Manager (A)

Date and time: December 8th 2022 12:00

Location: Virtual meeting via ZOOM

K: Record exactly/ So, oh, it's already running. Perfect yes super.

A: Perfectly wonderful. But I had also seen, are you also doing some self-employment in the area of hosting on the side?

K: Yes, that's the idea. So I started at Search One with a completely normal WordPress hosting comparison and then realised, hey, that's actually a cool topic and you'd have to do more, because as I said, the market is completely non-transparent. And next year I want to use the research work to program the benchmark. Then hopefully, when everything works, I will measure the hosters out there. So on the one hand, of course, the topic is performance, but the other topic is also support quality. I want that to be standardised, not via a script, of course, but via people with questionnaires and so on. Yes, I would bring in a bit of transparency and then once, yes, a kind of comparison portal, so to speak, the big hosting comparison/ But also for hosters, we can also talk about it again, I don't know if that is interesting for you. I have already spoken with some hosters who want to benchmark their own systems externally, so to

speaking, and also benchmark their own support again and again. So are we good, are our support staff good, are they competent, are they qualified and so on. And if I already do all that anyway for my comparison, I would then also offer it as a service, so to speak. And that's where I am now. First I have to finish my thesis and then I can start. I already have a few comrades-in-arms. I'm in the process of putting together a small group. But it's going to be a bootstrapped side project more or less.

A: Yeah, yeah, it'll be cool, good. Yes, interesting. Very interesting definitely.

K: I'll have to check, I know a bit about [EMPLOYER]. That's mainly infrastructure as a service, I know, but do you actually do hosting? You don't have any small products, do you?

A: It works, so with us you can do everything, but also nothing. So, we are actually, I'll say/ We see ourselves a bit as, well, we are actually a software as a service company, that was the basic idea. It actually developed from an idea that you equip data centres, no matter where they are, with a stack that comes from us. Directly on their own hardware and then, in the end, the whole system is under the management of [EMPLOYER]. But it is at the customer's site.

K: Oh, ok, so you have your own hardware, but you sort of take over the virtualisation and the, the management of the whole thing.

A: That's right, and not only that, but we also build automated services, so we build our services in such a way that they also work on the customer's hardware, but are under our management. That means, for example, if you think about a Daimler, let's say a Daimler that has its own hardware and says OK. That's my whole business logic that I have on it, that I don't want to have in the cloud, but actually wants to use all the advantages of a cloud, i.e. managed databases in Kubernetes and such, then he can take our stack, can license it and then has the possibility to use everything that you currently see with us in the public cloud area. To run it on their own platform.

K: Okay, okay, do you have a "best-of-breed" open source approach or did you really develop everything yourselves?

A: This is completely self-developed.

K: Okay, wow.

A: Yes, we are completely self-sufficient, we are not tied to anyone else, in other words everything comes from our own pen and everything is developed by our own team here, there are almost a hundred people here in Cologne. And from this, let's say MVP, that we say ok, we're going to do it this way, that we show how it could be, the [EMPLOYER] actually came into being, as you see it externally. So we started by offering hosting services, but you have to

differentiate a little bit. We don't do this very classic hosting, where you get a directory where you put five HTML files and then they are suddenly online.

K: With Assign Domain and then that's it and create a database.

A: Exactly right, exactly, but for us it's really about virtualising everything. In other words, we started by saying OK, we'll show you how it could look, that you don't virtualise a VM via a VM ware or something else and then in Proxmox, but that you just say OK, you have an interface that is somewhere on the Internet and if you click on it, something happens on the other side. So/ That led to us getting quite a lot of customers relatively quickly.

K: The whole thing is also web-based the management software?

A: Everything exactly.

K: And on the machine? So if I really arrive now with one with a "bare metal", do I first install an operating system? And then your/you actually go straight to the hardware?

A: Exactly, that means in the end you get from us what feels like a, not Raspberry Pi, but you get from us a small computer. You put it in your rack, wire it up and then we get IPMI access, which really means access to the hardware itself.

K: Oh, so you also need the Ethernet bios thing, the port and so on. So really, ok. I had to go through this fun once with an HP server. That was not so funny.

A: But the cool thing is that our logic is so cool that it can also say, for example, that if a hard drive breaks, our system automatically reads the serial number and makes an automated RMA.

K: Ah cool, ok yes ok, so you've sort of thought the issue through there.

A: No, we're definitely on the right track, I'd say. And we have, of course, we have other people, you know, like an Azure stack, that's also available, you can get it if you want, but you still have the problem that it's still Azure. So you still have the/

K: Yes, data protection and so on, that's all, yes.

A: Exactly, and when you talk about costs, it's much more expensive to get an Azure stack than, for example, in [EMPLOYER]. And you still have German support. For example, we have the company [CLIENT] in Switzerland. They were looking for a cloud offer. They came from a normal hosting company, wanted to have a cloud offer and were looking for a partner. Then they took a look at what we were doing and said, wow, that's so cool and we have a lot of hardware in Switzerland. And we also have hardware in Austria, so how would it be if we used your stack? So/

K: Ah, now I understand the claim [CLAIM]. Yes, ok/

- A: Yes, that's right. That's one of the things where we just, I'll say, where we come from and of course, over time and the more customers we have, they have also said that they would like to have managed database services. We don't really want to take care of our database ourselves anymore. We want someone to do it.
- K: And then you also started to build up your own hardware with the software and then ultimately sell the services as/
- A: That's exactly why we started to put our developers there, or even more developers, to build services such as Managed Kubernetes or, let's say, Managed SQL, a managed mySQL.
- K: You even build GPU instances and everything I have only seen.
- A: That is exactly yes/
- K: But that is then actually hardware that you operate to say, that is then already at the end of the day.
- A: At the end of the day, when you start a project with us, you have the choice between various data centres, which can be ours, but they can also be from our partners, because we built them that way.
- K: Oh, so everyone can sort of put their resources in there as well and oh, ok/
- A: If he wants to. We also have partners who don't want to appear at all. They just want to use our stack, but say we actually want to fly completely under everyone's radar. They don't do that, of course, but theoretically we have a brokerage model in there/ where you say hey, I still have 50 cores free.
- K: Yes, of course, you can change your capacity utilisation, yes, yes, yes.
- A: And a terrabyte of ram. Right on.
- K: Smart, ok, yes, exciting/ Yes, then I'm actually with you at the best possible point, so to speak, because this is your core business, which for many, I'll say for many hosting, when you talk to hosting companies, they often don't really have a clue. They just take something off-the-shelf, install something and don't know what the things do.
- A: Yes, right, right, yes, yes. And the problem is just the wording cloud is already like that/ So I see that in the conversations. I deal with customers who come from a [COMPETITOR] world, for example, where it's really like this: you get a directory somewhere on a hard drive and say you're in the cloud. But what we have is a real cloud. So with us it's just like this, for example. If you place or host your website with me, for example, I could never tell you exactly where it is within the data centre, because our software, for example, distributes the workloads automatically.

- K: That brings us to the exciting point. Exactly. So this dynamisation of resources also means that at the end of the day, it has to be limited somewhere, because you can't say, no matter how much demand the customer is currently drawing, he just has package XY, whatever you call it. When and how are things limited so that you end up with an overload, or? Yes/
- A: That's how it is with us, we don't have any packages, not at all. With us, everything is completely "pay as you go".
- K: I see, you have simply deducted the resources used, so to speak.
- A: Exactly, we have a pretty smart API behind it. It tells you exactly what resources you are using with us by the minute. And even if you don't use them anymore, you don't have any billing. So you pull up a VM, say ok, super cool, but I only need it during the day, because I only develop during the day, then you just turn it off at night.
- K: Okay, I see.
- A: Then it won't cost you any money at night.
- K: So these are/ Yes in the end just like a real yes/ With Amazon and Azure the billing models are similar, exactly.
- A: Exactly, and we have noticed with our customers that they are keen to continue using services or, let's say, things that they know from AWS or something like that in Germany. And the whole thing is coupled with, let's say, super modern services.
- K: Do you also have an API compatibility, so to speak, that you can just kick Amazon out and take you in, so to speak, without changing the code more or less?
- A: It always sounds easier than it is.
- K: Yeah, sure. But at the end of the day.
- A: Ultimately, we are completely API-first as far as everything is concerned. We have, let's say, the latest or most important things that are used for deployments like Terraform or something. We're officially in there, so there are quite normal Terraform providers for [EMPLOYER]. That means, if you are familiar with it and know how to use Terraform, for example, and now have, let's say, an S3 from AWS connected in addition and so on. Then it's only a few addresses on paper that you have to change. But it makes it very easy for everyone, I'll say, who really knows about Dev OpS in the cloud area, to work with us.
- K: OK. Yes, that's almost already/ Because at the moment when you can theoretically make unlimited resources available to the customer, but charge him for what he uses, that's also smart for you, because then you don't have to deal

with the resources in the same way. But let's assume I would now say that I would like to offer hosting for my customers on your infrastructure. And my customers expect me to pay them a fixed monthly fee/ I mean, sure, I have to, I have to, yes, in principle, cross-subsidise and calculate so that if one has a little more and one a little less, the whole thing pays off. But do you have the possibility to set limits at all? Or what are the options available to me?

A: The bottom line is that you have two options. One possibility is, let's say, you create an account with us for you, for you. And say, I don't know, for security reasons, I want to limit that. Then you can limit every single resource, CPU, RAM, storage, number of IP addresses. You can do that, let's say, to the amount of the resource. But you can also switch off the resource itself, that you say ok, I never want to get into the embarrassment of accidentally clicking a Kubernetes cluster with 18 nodes.

K: Okay, yeah, okay. Then it is simply not available at all.

A: Exactly, and that is one thing. And the second is that we are also very, very active in the reseller area, especially because of the technology that is underneath it, where it is important, I would say, to map a client structure that is as separate as possible. That you say I have 50 customers here who all run on [EMPLOYER], but they are 50 separate customers who ultimately appear as 50 invoice items on one invoice. Of course, we have that in there too, because that's where we come from.

K: Yes, then you can also assign it better. Sure, yes.

A: And you can go completely granular down to each customer. You can determine for each customer that they are allowed to use X amount of cores, for example, so that you can limit it a bit and accordingly also, I'll say, build a bit of a cost brake into it.

K: But what is a core? Is it a virtual core or is it a real core?

A: With us, they are real cores.

K: Okay so, it's actually then when I have a core, that core is exclusive, at that term only to me.

A: Yes.

K: And yes, well, that is the CPU that is currently installed in the server.

A: Which is built into some server. For example, because we have real cores and don't do any over-provisioning, you actually have huge advantages, because you don't have any CPU steal times or anything like that. You always have full power immediately, which is actually the same thing. And you simply have the power.

- K: How quickly does the system react? Let's say I have no idea, let's take a very stupid example, because it's so striking. I'm now on "Die Höhle der Löwen", but I don't tell you about it beforehand and suddenly a hundred thousand people show up. Like this. Now.
- A: We have a lot of good experience with that, especially with "Höhle der Löwen". Because there were a few cases where the hosting was with us. Where the servers held up for once and didn't go down via the others. But that depends a bit on the infrastructure that has been set up. So what you use, for example, if you use, I say /
- K: Well, if the bookings all end up accessing the same SQL and it has a lock or something. Then it is clear.
- A: It depends, so if you now, I say, if you say, ok I want to take care of the SQL myself, then you also have to determine the sizing yourself. But if you say, I don't want to take care of any peaks and I don't know what time of night, then you can just go and take our managed database.
- K: Ah okay, and that scales fully with it?
- A: Completely autoscaling.
- K: Is that quasi a Galera cluster or what is that at the core, quite likely, right?
- A: In essence, yes, it's different. It's just that with us, they are pulled up as micro-services and then scale in the end effect, like with a - let's say stupidly - like with a Docker container, which you can also restrict or enlarge theoretically.
- K: And what about replication and redundancy etc.? Have you/ How does that work then, if I now/ You can't then always write into the same container, so to speak, at some point it has to be merged again, right?
- A: At the moment, we don't have any real redundancy, let's say as we know it, three machines, three Docker containers. At the moment, that's a bit of a, yes, if I could decide, I would put it much higher on the priority list. That is also a topic we are working on. Nevertheless, that doesn't mean that we don't make backups ourselves all the time, even on the side.
- K: Okay. Yeah, you have to anyway.
- A: Exactly, firstly us and secondly, it's also something like that, we deal with it quite openly. With us, if you use a managed database, you are not in vendor lock-in. You can always make your own backups. At the moment, that's a bit of our recommendation in order to, I'll say, defuse the situation a bit. But we also have, let's say, quite large projects from the IT sector, where 1,000,000 hits land on it every day, easy, on the service, and they still use the managed service, because it's just great and works great.

K: Okay.

A: Exactly. In the end, you can limit all, let's say, all possibilities, because you have, let's say, the database running. It also performs great when there is at least little traffic and if you now have no idea and have placed an ad that goes live on Fridays or something and you have a super high traffic, a blatant peak, where in the worst case your database would normally crash or would become slow, you have [unintelligible] scaled up and then down again.

K: OK, now I have to sort myself out, because this is really a different conversation than with a classic hoster.

A: Sorry.

K: No, all good/

A: But that's exactly what I meant at the beginning. That's what distinguishes a [EMPLOYER] from a classic, let's say, hoster who somehow puts a rack somewhere for you.

K: Yes, I mean that the basic topic is then also already/ Well, I only know from customer experiences who then said, yes ok if I really/ So real cloud hosting, that was always the statement, real cloud hosting is so expensive that it is then often not worth so much to them to have this scaling option. Because the yes/ How are you priced there? Can you somehow/ do I know/

A: I have an AWS invoice lying next to me from a customer. Who now wants to come to us from AWS. Depending on what you use, we are actually always cheaper than AWS, for example. Because we also do things like, I don't know, calls on the S3, or the S3, which is just a huge topic, traffic on the S3, that's what AWS, Azure and Google earn a lot of money with, simply because it's not assessable.

K: Yes, of course, the storage space is free, and they'll be happy with the traffic.

A: Exactly, and when I look at this, it's a bill in the high four-digit range every month. And a huge part of that is traffic. And we have a fair usage policy. If you have an S3, it doesn't matter to me or to us whether you make one call a day or a hundred thousand. Mega. So it costs the same. That means we have a much, much higher cost transparency right from the start. But we're not the cheapest either, because the fact that we're in the cloud means that, for example, you won't get any storage from us that isn't designed for triple redundancy. We did this for one simple reason: because the customers who work with us and who want real cloud hosting have no problem with the costs. They don't. We have a fairly large manufacturer of kitchen machines here and he says, "Cloud Connected Devices", and he says, "I have a few million devices here that are on the road worldwide."

- K: It must run reliably and it must not run on one of the US providers.
- A: Exactly right exactly and/
- K: Yes, okay, that's a completely different use case yes/
- A: Exactly right, and we always have the problem that when we are up against, I'll put it this way, when he's up against a [COMPETITOR], for example, we will always lose in terms of price, if the customer's goal is to put five HTML files somewhere, we will always lose. But that is not our claim. In the end, our claim is good performance, great support at eye level and the bottom line is that you can scale with us. Whether you do it, your problem, your use case, I say. There are also customers who scale zero with us. They have their workloads, which always remain the same. But we also have some who have no idea about Kubernetes clusters, who scale up and down all the time, but in the end that's up to the customer. If I had to say how we see ourselves compared to the other German competitors, I would say that we are always a bit more expensive than [COMPETITOR], for example. But we also have different services than [COMPETITOR]. And I would say that many little things make us stand out. We will soon have a system that will make it very easy to build third-party managed services. For example, which is really cool, because the customers who come to us come from an AWS world, an Azure world, a Google world. They are used to being completely full up to the ceiling with some kind of managed services. They no longer want to rush into any Linux systems, they just want to get as close as possible to serverless. That's just the clientele we work with.
- K: Okay. At the end of the day, ideally, you don't really need your own Dev Ops team any more, do you? So yes, at least no more hardware team, so to speak?
- A: Exactly. You notice this extremely strongly in our system houses, for example. Because we have this reseller model, we have a lot of system houses that, when they talk to their customers, talk about the cloud or buy new hardware. A make-or-buy decision. And that is very often, I would say, a calculation. But it's only a calculation until you start to calculate what it costs for the support technician to get into a car at night, drive 200 kilometres north with a hard drive in his hand and somehow screw the thing in, try to get it going again, and in the meantime the whole machine stops. Like this. Then the cloud pays for itself very quickly.
- K: Yes, that's right. So that's right, if you then have any SLAs in there that really also have short-term response times etc. then yes, I completely agree with you.
- A: Yes, and that's when you realise that a large part of it, I'd say, simply falls away. This naturally stirs up a bit of fear, especially in system houses, because the employee who actually thought it was cool to earn a lot of money because he

had to go out again in the afternoon or at night is naturally afraid of becoming a bit unemployed.

K: Yes, but hopefully he'll get other tasks again, so I think/

A: Exactly. And you notice it with the customers who use our stack. They just say, ok, what we gain from this, because we are simply ultra-modern all at once, is worth much more.

K: Yes well, you also attract a completely different talent when you are then on Kubernetes or so on and or have previously piddled around there with your own. (laughter) Whether that's true, that's true. [unintelligible]

A: Yes, totally. These are partly / Well, we have a customer who is actually so dusty that you say, wow, it's a miracle that he's even starting to move in the direction of the cloud. But he also has someone sitting there who says, hey, I don't want to take care of it any more, so that it runs somehow. I want it to run. I want it to be super-performant, no matter when, and then we'll just make a hybrid cloud thing out of it. Then I'll just say that the big treasure remains in the basement, stupidly put. And all the services that we don't necessarily need in-house, hey, I'll take care of them.

K: So everything out/ yeah, ok sure. Also smart, yes.

A: And that works with us. So you can't, when you log in now, within, I'll say 10 minutes, you could create a project in Frankfurt, which is located in our data centre, and you create a project in Amsterdam, which we've had live since this week. Then you have two completely separate locations. And if you're up for it and you have hardware flying around, then you have another stack, then you have the third one at your place, let's say in Berlin. Then you suddenly have three locations within a few minutes. Super cool. Yes, and that is of course something that depends a lot on the target group. I used to have these conversations every day. Some say yes, ok, we see what's involved and what that implies, a change to the cloud and we also see what sets you apart from a [COMPETITOR], an [COMPETITOR 2] or something like that. We also see that and at the end of the day the question is always/

K: [COMPETITOR 2] also builds/has built quite a large cloud infrastructure. I think they are also serious about this.

A: Definitely. You don't hear, I mean, I'm not saying anything bad about competitors. Never.

K: Yes, all good.

A: It's just a different way. That's ok and quite honestly/

K: Everyone has their own approach, and that's OK.

- A: That's right, and our USP is simply different. And we don't have the company behind it, like 1&1, which can just pump money into it. It's different for us, that's why we actually do it in line with the market. Only that we say ok, we listen to our customers. So with our customers, we always have an open ear. So we have projects/ We have a customer who is active in the industrial sector in the medical sector and he just said, Hey Alex, pay attention. He's super. I just have a problem, I actually need a way to be able to audit properly. We know this from AWS. We've always had that, audit trails and the like. For him, we simply upgraded our Postgres and built in audit trails, our own, something of our own. It just took a month or two. But the customer is just ultra-happy and tries that with one of the big ones, right?
- K: Of course, you don't get any individual solutions there, that's clear, yes.
- A: We just do small and medium-sized businesses, right? So, these are the customers who understand us.
- K: But then? I think I have an idea now, then I might come back to you briefly, if I may with a follow-up, but we can do that. Then do it by e-mail.
- A: I'd love to.
- K: Because in the introduction to the thesis I would then compare your model with the usual market model in the hosting sector/ Because the problem is, I'm approaching from the bottom up from the/ I'm saying quite deliberately, so to speak, here I want a hosting package for 20€ a month. How can resources be sensibly managed, monitored and allocated? And to what extent can I perhaps do that as a customer? At the end of the day, I want to have comparability. You are at the level in inverted commas, you don't even compare yourselves, because you also say you get as many resources as you want, just pay fair use, I think that's great, but it's a completely different approach. Then I would really put the approach against that and then just add to it and yes such a kind of comparison/ The topic of cloud hosting is also something that I think will generally become more and more in the future. But that also means, if you were to advise me now and I now say I don't know, I want the next one here, such a small one, I don't know, do you know [COMPETITOR] or such a [COMPETITOR] or so/ So and I would say, I now open up such a WordPress specific hosting provider. Would you then tell me privately, better leave [EMPLOYER] alone, because if you have customers who really have a lot of demand, then it will also be expensive. And then you'd better build it somehow else, or would you say even for such a case it could work really well for you?
- A: Ahem
- K: So there are many advantages that you don't actually need, that you offer, so to speak.

- A: The point is that we now also have a way to become considerably cheaper. Since this week, we have a location in Amsterdam, which is still GDPR compliant, or at least close to DSGVO. We have it for this very reason, so that we can also serve those customers who say ok, I have/ I am in a price-sensitive area. Where perhaps customers are, I would say, lower to middle middle class, who say ok, we see that we have to change something, we see that we need certain things, but we don't actually want to pay this price. That's why we have the data centre in Amsterdam, where we can now also deploy workloads, i.e. the customer. This means that theoretically you can now run a two-stage model, that you say ok, you get, I'll just say "[EMPLOYER] light". I just call it that. And then you just get [EMPLOYER], like this. At the end of the day, you can have such a wide range of customers, especially in the area of WordPress and Co.
- K: Yes, well, sure. There are huge sites that run with WP. That's already clear, yes.
- A: Exactly. Basically, I would always say, yes, is a client for us. Always. If he wants his website to be up and running, to be fast. If there is any problem, that he gets good support. And that he has the possibility, if he gets a certain growth and thus has to scale, at some point, that he is not stuck in some package. Or to put it very harshly: If the thing just goes down the drain and he has to close down or something or just put it on ice, we had that very often during the Corona time. With us, you switch off the server, leave the storage on, you suddenly have only a fifth of the costs and when you feel like it again, you switch it back on. Done, no. If these points are even remotely important to you, then you are a customer for [EMPLOYER].
- K: Ok.
- A: If you say you'll put the price up. Then don't.
- K: Yes, but that's us/ That's the exciting thing about hosting. I've really heard from those who, let's say, have seen the [COMPETITOR] advertisements on TV, hosting from 1€ and then they also want hosting for 1€.
- A: Yes, that's right.
- K: Until then/
- A: But then they also get hosting for 1 €, but/
- K: Yeah, it's like that. Yes, oh God, there are also stories, hey dude. Do you ever/ No, no, don't do it, you'll only get a headache. Leave it, leave it. But next year I'll have to make customer accounts with all the hosters, so to speak, in order to test them all. I'm already looking forward to it.
- A: Wow, that's going to be [unintelligible]. So then I would think strategically about when you do one with us, because I can tell you from my own experience, I've been through a few hosters because of my agency experience.

- K: The problem is that I don't have to do it for myself in order to find the right hoster for me, but I have to do it for everyone in order to establish comparability, which doesn't help.
- A: Hhm, so our / So the customers I talk to, they come from either an AWS world or just I say from a world where there's just a Plesk behind it, and just when you become a customer, you actually get a Plesk instance.
- K: Exactly the Plesk world, that's just the main case, I think at the moment.
- A: Yes, definitely. And the people, they love us, right. They come to us. They say, wow, I've had a look at your place. Hey, that's so cool, it's so cool, because you see immediately what it costs, you have sliders and see ok, that costs me x a month, and that's also x and not y all at once. Everything just works the way you would want it to. That is definitely an experience for many of our customers, who say it's amazing that it can really be done so easily. Because we're talking about ultra-highly complex architectures in some cases, which you can find here in the backend.
- K: Yes, of course, by clicking quasi to you / Yes.
- A: A Kubernetes cluster is ultra complex normal, right. With us, it's a managed Kubernetes. You don't pay us for the management, just for the resources. Like this. We have a completely unmodified Kubernetes, so it's as "vanilla" as it gets.
- K: Okay.
- A: In other words, we don't even have a metrics server installed, so there's nothing on it. That means you can do everything you know from another world and, at the end of the day, every tutorial you can read somewhere on the subject of Kubernetes, which is not from Lenote or any other provider, you can use one to one with us. And that's what people celebrate in our case.
- K: Yes, I believe that, I really believe that. But that's really the customer case when you run your own services or develop them yourself. That's not the standard, I want a standard CMS and I just want to build my website around it or click.
- A: So we have for I say for such standards, so we have for ultra-standards, such static HTML or so, right for example, there are still customers who find that good, right.
- K: Yes, of course, sometimes it even makes sense for security reasons.
- A: Right, for them we have simply created the possibility in our S3, which is also known from AWS, that you can simply put an S3 bucket live. Then you just route an IP address to it and then you have your static hosting, which costs you nothing.

- K: Then you just pay for the traffic, so to speak.
- A: No, to be honest, you really only pay for the bucket, i.e. what's in it. So if you just say/
- K: Yes, a minute price or a size price, right?
- A: Exactly, so I'll say quite, quite stupidly that S3 storage costs €0.05 per GB per month if you have it in Frankfurt. Like this.
- K: Oh, so cheap?
- A: And it doesn't matter if you have a call on it.
- K: What?
- A: Or whether this is a website that has really blatant traffic.
- K: Yeah, wait a minute/
- A: Wait a minute/
- K: That means that if someone were smart enough to keep the dynamic part of the website as small as possible, so to speak, and to always calculate and deliver most of it statically, you could very well host it very, very cheaply.
- A: Yes, you have halt/ The thing is, you mustn't forget, you have things that lie behind it that you know from systems like [COMPETITOR], for example, as far as e-mail servers and such are concerned. You don't have that there, of course. It's really like that/ For me, for example, I tinker a bit with the Hugo Framework privately, which at the end of the day throws out a static HTML. So I put it in a bucket and set this bucket to public and then refer to it. Theoretically, you can copy this for as little as €0.
- K: Yes, there is a similar model or a similar trick in the Google Cloud. I don't know, it's been a while since I read that, but the bottom line is that you always let the CPU instance sleep and almost always only deliver code, static code. But at the end of the day, it's always Google, but I mean, most people have their email server somewhere else anyway, hopefully. It's always a big issue just because of email archiving and so on. You probably don't do that, do you? GoDB-compliant and the whole rat's tail that goes with it.
- A: We just have partners who do it with us. For example, we have just/
- K: So ok, they'll have their service on yours, so to speak, yeah ok.
- A: Exactly. So we are actually also a very, very popular partner for, let's say, German SaaS companies or, let's say, SaaS companies based in Germany,
- K: Yes, I think so.

- A: When it comes to document management, audit-proof filing of any kind of stuff we have/
- K: They then run their own application on it, so to speak, which does the whole thing.
- A: Exactly, that's what we have in the end, I'd say the software solution is, I'd say, compliant in itself, because it stores in an audit-proof manner, for example. There is a [EMPLOYER] underneath that works per se in compliance with the GDPR with all the necessary certificates and then you have, for example, your assets in S3 with us, which is then set to versioning with one click, so that it is also versioned in itself. Great, isn't it?
- K: So you abstract away their hardware layer, so to speak?
- A: Exactly right. Exactly, they just take care of what they can do and that is ultimately to develop the application further, right. And we just do the rest. So that's also possible, and yes, this topic in particular can only be done with static websites, that's something we don't have at all, so sure, we've already thought about that, but I don't know many customers who really use that.
- K: No, this is not the first use case either. You guys are really about application hosting mainly. That's clear, yes.
- A: Right, exactly.
- K: That was just, when I hear something like that, the gigabyte costs 5 cents a month, no matter how many accesses and so then yes, of course, the imagination goes. But it has nothing more to do with the BA or the Master's thesis, I would say. I would like to keep you in mind, on the one hand, when it comes to hosting our actual applications, because we are also going to build a WordPress plugin with which you can simply measure the hoster you are with.
- A: Mmm, cool.
- K: And then all this data has to get back to us somewhere and I think I'll keep an eye on you, because this is something where I don't know yet whether 100 people will do it, 1,000 people will do it, 100,000 people will do it. So you know, the WordPress community is big and when something goes down, it can really go down.
- A: Yes, that's true.
- K: Super exciting.

A.4 Interviewee 4: Nip & Tuc

Interviewer: Kai Spriestersbach (K)

Interviewee 1: SVP Product (N)

Interviewee 2: Software Architect & Technical Product Owner (T)

Date and time: January 11th 2023 14:00

Location: Virtual meeting via Google Meet

N: Okay, let's start at the beginning. Best of Breed. We don't do that.

K: That's what I thought, yes.

N: The platform has grown over 25 years. There are a lot of in-house developments. If you look at it, the hosting machine as such is first of all sheet metal. However, this sheet metal is virtualised to a certain extent. For us, it's just LXC containers for now. We introduced this a few years ago, simply to be able to manage the waste better. But it's not virtualisation yet. So in principle, the web hosting stack runs on a sheet of metal, just containerised and the whole thing is, as far as our platform is concerned, an Apache, not off the shelf either. There are a few modules of our own in it, for example we don't have something like vhosts, we have a specific Docroot mapping. We also have a specific changeroot mapping for everything that is PHP execution. That means that everything is designed for mass, so that ideally you can control it via databases or the new platform, via a Redis database, via Redis mappings, and of course you can update it relatively quickly. This means, for example, that compared to a classic Apache, you simply don't have to restart your web server. It simply always runs. And adding or removing a domain binding is relatively easy, because the Apache module that we built for this purpose takes it directly from this mapping. That means we have a lot of in-house development. As far as gear redundancy is concerned, we also have our own development in the kernel. We have our own block-level replication, which means that each piece of information is always available twice in two different data centres. Of course, this doesn't have that much influence on the performance of the benchmark. But it is definitely relevant, because when something happens like at OVH last year, you notice that it's good. And of course it also has the advantage for the operational side that you can always take a box down, wait, because it can swivel at any time. And that is definitely an advantage as far as that is concerned. As far as performance limitation is concerned, there are of course a few limits that you set as a U-Limited Centre [unintelligible] etc., simply for pure protection. But also the database has such limits, that is the regular database has I think ten parallel sessions. The performance database would then have 25, I think. I wouldn't let myself be pinned down here, we'll have to ask again.

- K: Yes, that is not relevant either.
- N: [unintelligible] But of course there are. But that's more for the protection of this platform. As far as the limitation towards the customer is concerned, we have also built our own. Namely, where we do the PHP execution, we have something like our performance level. And that means quite concretely. On the one hand, it has an effect on the memory that the process can use as such and higher performance levels also have more per executor. But in principle, an execution pool, that is, a current performance level may have 15 parallel executions. That means 15 parallel PHP processes and then that's a soft limit, which means that we still cue slightly above that. That means we will not reject it yet. We also have statistics on this, which the customer gets directly. And if you go beyond that, we will of course have to refuse, which means that the customer will have to book more at that point. Of course, there are also ideas to make this more dynamic, but that is not yet the case. But that is the limitation we have. That means that parallelisation is relatively dependent on how high the respective performance level is. Behind each performance level there are basically a number of executors. Very small tariffs, extremely cheap tariffs sometimes have only two. But you also have to say that two parallel executors can serve a relatively large amount of web traffic.
- K: Is there some kind of caching mechanism before the actual execution, so that things that have to be delivered statically, for example, are also cached? Or do they all run through the same pipeline, so to speak, every request?
- N: At the moment, everything still runs through the same pipeline for the Word-Presses, because it is PHP. We have already optimised performance by using a caching plugin for our managed WordPress sites. That means that in principle they are static sites. So, the next logical step would of course be to say that they are static sites. That means that everything that is static goes through. But that would be a problem, of course. At the moment, WordPress still decides what is static and what is dynamic. And I think that's probably a bigger sticking point. Where it could be decided outside, it would of course be simpler.
- K: Yes, absolutely, yes good/
- N: Static content as such, i.e. graphics etc. are of course delivered directly. They do not go through the execution at all.
- K: Okay, yes, that's exactly what I was referring to. I didn't mean caching caches or things like that that are kept in the WordPress system, but real, yes, an image file or a CSS without parameters or something, that is simply hidden, so to speak.
- N: They are left out. They don't cost anything either.
- K: Exactly. Okay.

- N: There is a difference when you start up PHP Interpreter and have to interpret on a large scale with PHP, or of course we have an OP cache running there too, because compilation is the most expensive thing.
- T: Maybe there's another layer coming out, because you somehow addressed CloudLinux and Plesk specifically. Of course, we also took a closer look at CloudLinux. Not so long ago, about a year ago, because we were also talking to them. And we haven't found anything that has more competence than what we have built up over the last 25 years with our own Linux kernel developers. We even compile our Linux ourselves, so that we really adapt the Linux to the platform in such a way that it is ideally designed for our usage scenarios/ In essence, it is virtually the operation of some CMS and here, of course, WordPress is at the top of the list. That's one thing. And the other is Plesk, which is more of a life-cycle management platform for the individual WordPress instances. Here, too, we have our own life-cycle management tool. So we have built a life-cycle ourselves, how we not only deploy the WordPress, but also keep it up to date, how we do the plugin orchestration for the delivery. You always intervene when things get too abstract.
- K: No, it's all good.
- T: I'll have to be a bit more specific, but we don't rely on third-party solution providers here, but actually do it with a maximum depth of added value.
- K: I almost thought that you didn't want to be dependent on licence models etc. in Scale either. Plesk is also making great strides in this area. And I think that will get worse in the next few years.
- T: So even there, when you talk a bit out of the sewing box. This year in particular, we've noticed that because things have become a bit tighter in the economy, all the third-party software providers have come here first and held out their hands wherever they can. And when we talk about core business, of course, and hosting/WordPress hosting is our core business, you can't make yourself dependent on third-party providers. Because the market in particular is so highly dynamic. Plesk was bought by the WebPROs not so long ago.
- K: Right.
- T: Are then also a private equity in the back/
- K: and you don't really have a choice anymore if you want to have a commercially supported management system.
- T: Exactly. But there is also private equity behind it. They have only one goal: to increase the value of the company. And then comes the next and the next and the next. And if you are a provider of end-customer solutions, you can either

make it more and more expensive for the end-customer or switch from BUY to MAKE right from the start and do it as early as possible.

N: So you don't have to do everything yourself. We also have an installatron in the WordPress section. However, he also has a lot of custom code in there and in case of doubt, that's a cheap solution for us at the moment. But I wouldn't say that we are dependent on it. That is, if it were to somehow no longer run well there, we could also change it at any time.

K: Alternatives, also in open source probably or what/ Or do you then develop parallel solutions quasi as fallbacks or?

N: So once in doubt, in the simplest case, you could take that at WPC-Lean [unintelligible] and take an Ensible playbook if necessary. You can also carry out an installation or upgrades with it. So you are already flexible, as long as there is a tooling and you don't completely lock yourself in, everything is fine. And at the moment Installatron is a solution. It was chosen once, and it's relatively good, especially for other applications like our One-Clicks and Click-and-Build, because we used to do that ourselves. We maintained analogues and application catalogues and packaged applications. That's just work if you don't have to do it, and as long as it somehow comes at reasonably favourable conditions, you can of course just buy it in. Moreover, it is expandable, it has open interfaces. That's not necessarily the most beautiful thing, but we have our extensions in it and it remains ours.

K: Okay. Great. I just have to ask again for more specifics before we come back to the topic of benchmarking. You mentioned earlier that there is some kind of containerisation running on the actual bare machine. Do you also have a pooling of the/ So before the VMs are created so to speak a pooling where several actual hardware boxes are taken together which are then shared again? Or do you actually have rack after rack after rack always limited to one hardware machine, which then has to be migrated or transferred, or is there a kind of, yes, complete virtualisation of the entire infrastructure?

N: It's still sharding at the moment, as we had it 25 years ago, so it's individual hosting machines. In the past it was really one-to-one. So we have the classic hosting machine, so to speak. In the meantime, what you see logically as a hosting machine is a container that runs on these physical machines, because they are simply much bigger and historical machines/customers are bound to their machine for the time being. This has advantages and disadvantages. Such a move always means copying and synchronising. Moving containers is relatively easy with our Mars platform. That means it is also possible to move remote block storage in and migrate mail over. So we already have tooling and of course our own tooling in our kernel. However, yes, it is a sharding approach, which also has the advantage that if one machine fails, only one machine is

down and the other 2000 continue to run. Which is also a risk mitigation at this point.

- K: Yes, of course. On the other hand, of course, you can't respond as flexibly to individual/ So now you have this kind of inflation of a single customer, it's really limited to the hardware of the current host. But it probably makes sense in this model. You don't charge according to consumption. That is quite normal over-selling, over-provisioning, I suppose. That you look at an average utilisation that you have on each box or that it runs stably.
- N: Exactly. We have tools for this that basically take over the allocation. We also have a few security mechanisms, for example, the quota cannot be written directly ad hoc at once, but must be done step by step. You will also see that there is an increase, so only one customer can say, I want to migrate everything immediately. Then you get the thing set up straight away. However, as a rule, one grows naturally and such an ad-hoc increase, especially with a 100 gigabyte space, is not a loud motive anyway. It's more of an abuse case. In this respect, of course, we also have brakes in there and this helps us to say okay, if the storage becomes scarce, you may also have to move containers and relieve the systems. And of course there is a bit of operational effort involved, that's for sure. It has advantages and disadvantages. The advantages are really partly the costs, because when you move towards a big cluster of this size, it becomes much more expensive at some point. And if I remember discussions with our kernel developer, yes, he proved to me mathematically that a cluster of a certain size is always broken. There are also advantages to being a little more decentralised.
- K: Okay. But that's yes/ Okay. That also makes a lot of things easier. That means that you effectively run a single customer account, so if I really go down to the individual customer account, it runs on the same, constantly on the same CPU with the same physical RAM, so it doesn't bounce around between the machines, so to speak. You also don't have any network storage, so to speak, system or database system outsourced, everything is on one physical machine in the end.
- N: Database runs separately. The database is an extra database cluster. Of course, it's relatively close to it on the network side. We used to have it locally, but it's more practical to have it externally. Of course/
- K: But in terms of redundancy, it's probably not much of a difference any more, I suppose, is it? Nowadays with the available network technology.
- N: No, the network is simply fast these days. We used to have network storage, too, so we also used to have blade centres with network storage behind them. That also had disadvantages. As I said, the current setup is sharding. Individual machines that manage more logical machines, if necessary, and each machine is available as a pair, so to speak. That means there are always two data centres.

It is replicated one-to-one. So it is also possible to swivel at any time. And for short load peaks, you can also swivel individual containers over to the other side if necessary, which may of course be at the expense of redundancy in the short term, but still delivers performance. Of course, this must then be resolved again, because one/

K: Yes, well, otherwise everyone runs off somewhere in isolation.

N: Otherwise, redundancy will have been reduced to absurdity at some point.

K: Yes, that's right, that's right. Okay, great, thank you very much. Ultimately, if you define, let's say, performance levels yourself, do product development, you probably have to do some kind of performance measurement to know, let's say, at the end of the day, what it delivers and or in relation to it. I don't know if you benchmark your competitors or benchmark yourselves in comparison to your competitors. What kind of tools do you use?

N: Internally, we have a Performance Index as a Service. These are various values that come from monitoring, that come from the machines. The utilisation of the systems, health, etc. are then monitored or also monitored, what is the performance, how long does the delivery take, so in principle our view. And in parallel, of course, there are various external benchmarks that are used. I mean, on the one hand inevitably, because some of them are also published. On the other hand, there are colleagues at our product, I think they all have accounts, to carry out tests directly or somehow Google PageRank etc. is used again.

T: We have built a small Kibana dashboard internally, where we also have competitors on it, and we use the Google Lighthouse Score to break it down into the individual categories of the Google Lighthouse Score, so that we can use test installations that are set up in the same way as ours, and then also directly in the tabular benchmark.

K: Exciting. That is also the approach/ I would also make a combination of actually measured performance synthetically and this external view. That is, you also have a test instance that is identical and that you can then use to establish comparability, so to speak, okay, and then probably the respective APIs/KPIs? that are asked by the customers/.

T: Yes, the KPIs are not what we put on the outside. That's not the first thing we write on our website, whether it's time-to-first-byte or first-visible-image or these individual sub-KPIs that are aggregated in the Lighthouse Score. It's really more to kind of map out internally where we stand as well. We also have our WordPress platform at different levels in the individual brands. At [BRAND 1], for example, we have the latest iteration, I would say, of our WordPress instance running, and at [BRAND 2] we have an older one, so you can actually see just the comparison between old and new. Where, what has changed there and how

is the competition. What's always a bit complicated is when you think about performance benchmarking/ That's of course very artificial, if what we do is a similar instance for all competitors and for us, because of course there are also caching plug-ins, optimisation plug-ins that you can incorporate into instances elsewhere. CDN, something that boosts your indicators, but which is not the native platform performance, but again via a third-party caching CDN, which is always guaranteed. That's why/ Well, it's difficult, because you're very often comparing apples and oranges when you look at it from the outside. But from the user's perspective it doesn't matter, because he says, well, the main thing is that the thing is fast. And whether it's done via the original platform or via another caching tool that comes from the outside. Or maybe I pay another €100 a year or €150 for some caching tool. That's where it gets a bit difficult with natural performance measurement.

- K: So that is precisely my goal, to create transparency, ultimately in the commercial version. Then, when the thing actually goes into use, to also make a distinction between cached and uncached and to transparently disclose "Hey, this is now simply on the CDN or this came from the static cache, or this is really a query that was just assembled in real time by PHP and MySQL. There are also cache busting or workarounds or simply for example something like benchmarking the admin ajax in the backend, so to speak, because I know that very many, from my own practice, work with a lot of caching. For example [COMPETITOR 2] is unspeakably slow in the backend and in the front they are super fast. That is such a classic. That's why it will be a combined score of many, many different values, so to speak. I think that makes the most sense. That is, but you don't push a particular plug-in for a customer, so to speak. Some do, [COMPETITOR 1] for example, provides a WP Rocket licence for all customers, which is then activated with the standard config. You don't do that. You sort of leave it up to the customer what caching solution they want, don't you?
- T: In essence, yes. So, yes, you could say, in essence, yes. As Christian said, we now have our own caching plugin, because we will also expand this further in the next iteration stages. That means that our aim is to provide a turnkey WordPress instance so that the customer doesn't need a third-party plugin, and since we also programme for ourselves that we are European providers and want to keep the data in Europe, it's not that easy at the moment to get a cool caching plugin that isn't channelled through some US provider or some US server.
- K: That also means that you want to map the topic of CDN in your infrastructure, so to speak, or how far does that go, so to speak? So/
- T: Yes, at the moment we also offer Cloudflare as a CDN in our hosting environment and also Cloudflare licences there. But in the long term, we are also

looking for a European solution. CDN is a bit more difficult, because you really need infrastructure at relevant nodes to be able to cache in and deliver quickly. That means the investment is more than just writing code, but you also really have to put sheet metal somewhere. We haven't finished thinking about this step yet, but we want to build up our CDN infrastructure ourselves.

Appendix B

Source Codes

B.1 PHP Benchmark Performance Script

```
<?php
/*
#####
#           PHP Benchmark Performance Script           #
#           (c) 2010 Code24 BV                         #
#                                                     #
# Author      : Alessandro Torrisi                     #
# Company     : Code24 BV, The Netherlands            #
# Date        : July 31, 2010                          #
# version     : 1.0                                    #
# License     : Creative Commons CC-BY license         #
# Website     : http://www.php-benchmark-script.com    #
#                                                     #
#####
*/

function test_Math($count = 140000) {
    $time_start = microtime(true);
    $mathFunctions = array("abs", "acos", "asin", "atan", "bindec", "floor", "exp", "sin", "tan", "pi", "
        is_finite", "is_nan", "sqrt");
    foreach ($mathFunctions as $key => $function) {
        if (!function_exists($function)) unset($mathFunctions[$key]);
    }
    for ($i=0; $i < $count; $i++) {
        foreach ($mathFunctions as $function) {
            $r = call_user_func_array($function, array($i));
        }
    }
    return number_format(microtime(true) - $time_start, 3);
}

function test_StringManipulation($count = 130000) {
    $time_start = microtime(true);
    $stringFunctions = array("addslashes", "chunk_split", "metaphone", "strip_tags", "md5", "sha1", "
        strtoupper", "strtolower", "strrev", "strlen", "soundex", "ord");
    foreach ($stringFunctions as $key => $function) {
        if (!function_exists($function)) unset($stringFunctions[$key]);
    }
    $string = "the quick brown fox jumps over the lazy dog";
    for ($i=0; $i < $count; $i++) {
        foreach ($stringFunctions as $function) {
            $r = call_user_func_array($function, array($string));
        }
    }
    return number_format(microtime(true) - $time_start, 3);
}

function test_Loops($count = 19000000) {
    $time_start = microtime(true);
    for($i = 0; $i < $count; ++$i);
    $i = 0; while($i < $count) ++$i;
    return number_format(microtime(true) - $time_start, 3);
}
```

```

function test_IfElse($count = 9000000) {
    $time_start = microtime(true);
    for ($i=0; $i < $count; $i++) {
        if ($i == -1) {
        } elseif ($i == -2) {
        } else if ($i == -3) {
        }
    }
    return number_format(microtime(true) - $time_start, 3);
}

$total = 0;
$functions = get_defined_functions();
$line = str_pad("-", 38, "-");
echo "<pre>$line\n|".str_pad("PHP BENCHMARK SCRIPT", 36, " ", STR_PAD_BOTH)."| \n$line\nStart : ".date("Y-m
-d H:i:s")."\nServer : {$_SERVER['SERVER_NAME']}@{$_SERVER['SERVER_ADDR']}\nPHP version : ".
    PHP_VERSION."\nPlatform : ".PHP_OS. "\n$line\n";
foreach ($functions['user'] as $user) {
    if (preg_match('/^test_/', $user)) {
        $total += $result = $user();
        echo str_pad($user, 25) . " : " . $result . " sec.\n";
    }
}
echo str_pad("-", 38, "-") . "\n" . str_pad("Total time:", 25) . " : " . $total . " sec.</pre>";

?>

```

Bibliography

- [1] M. Seltzer, D. Krinsky, K. Smith, and X. Zhang, "The case for application-specific benchmarking," in *Proceedings of the Seventh Workshop on Hot Topics in Operating Systems*, 1999, pp. 102–107. DOI: 10.1109/HOTOS.1999.798385.
- [2] M. Andreolini, V. Cardellini, and M. Colajanni, "Benchmarking models and tools for distributed web-server systems," in *Performance 2002*, M. C. Calzarossa and S. Tucci, Eds., vol. 2459, Berlin, Heidelberg: Springer-Verlag, 2002, pp. 208–235.
- [3] K. S. Bhutta and F. Huq, "Supplier selection problem: A comparison of the total cost of ownership and analytic hierarchy process approaches," *Supply Chain Management: An International Journal*, vol. 7, no. 3, pp. 126–135, Jan. 2002, ISSN: 1359-8546. DOI: 10.1108/13598540210436586. [Online]. Available: <https://doi.org/10.1108/13598540210436586>.
- [4] A. Bogner, B. Littig, and W. Menz, Eds., *Das Experteninterview*. VS Verlag für Sozialwissenschaften, 2002. DOI: 10.1007/978-3-322-93270-9. [Online]. Available: <https://doi.org/10.1007/978-3-322-93270-9>.
- [5] D. F. Galletta, R. Henry, S. McCoy, and P. Polak, "Web site delays: How tolerant are users?" *Journal of the Association for Information Systems*, vol. 5, no. 1, pp. 1–28, 2004.
- [6] R. T. Douglass, M. Little, and J. W. Smith, *Building Online Communities with Drupal, phpBB, and WordPress*, en, 1st ed. Berlin, Germany: APress, Dec. 2005.
- [7] T. O'Reilly, *What is web 2.0*, Sep. 2005. [Online]. Available: <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html> (visited on 10/15/2022).
- [8] J. P. Casazza, M. Greenfield, and K. Shi, "Redefining server performance characterization for virtualization benchmarking," *Intel Technology Journal*, vol. 10, no. 3, 2006.
- [9] M. S. Thompson and S. Thompson, "Pricing in a market without apparent horizontal differentiation: Evidence from web hosting services," *Economics of Innovation and New Technology*, vol. 15, no. 7, pp. 649–663, Oct. 2006. DOI: 10.1080/10438590500418968. [Online]. Available: <https://doi.org/10.1080/10438590500418968>.
- [10] M. M. Hugue, *Different types of benchmarks: Based on section 1.5: Measuring and reporting performance, and section 1.8: Fallacies and pitfalls of the prescribed text book for cmisc 411, computer architecture: A quantitative approach*, 2008. [Online].

- Available: <https://www.cs.umd.edu/~meesh/cmsc411/website/projects/morebenchmarks/types.html> (visited on 09/29/2022).
- [11] pingdom, *Web hosting now vs 10 years ago*, Feb. 2008. [Online]. Available: <https://www.pingdom.com/blog/web-hosting-now-vs-10-years-ago/> (visited on 10/14/2022).
 - [12] M. S. Thompson and S. Thompson, "Intra-industry differences in vertical integration, heterogeneous costs and pricing: The case of web hosting," *Empirica*, vol. 35, no. 5, pp. 503–523, Jun. 2008. DOI: 10.1007/s10663-008-9070-7. [Online]. Available: <https://doi.org/10.1007/s10663-008-9070-7>.
 - [13] E. van Baaren, "Wikibench: A distributed, wikipedia based web application benchmark," M.S. thesis, VU University Amsterdam, 2009.
 - [14] S. Bose, P. Mishra, P. Sethuraman, and R. Taheri, "Benchmarking database performance in a virtual environment," in *Performance Evaluation and Benchmarking: Revised Selected Papers*, vol. 5895, Lyon, France: TPCTC, 2009, pp. 167–182. DOI: 10.1007/978-3-642-10424-4\textunderscore13.
 - [15] R. L. Grossman, "The case for cloud computing," *IT Professional*, vol. 11, no. 2, pp. 23–27, 2009. DOI: 10.1109/MITP.2009.40.
 - [16] J. Hoopes, "Chapter 1 - an introduction to virtualization," in *Virtualization for Security*, J. Hoopes, Ed., Boston: Syngress, 2009, pp. 1–43, ISBN: 978-1-59749-305-5. DOI: <https://doi.org/10.1016/B978-1-59749-305-5.00001-3>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9781597493055000013>.
 - [17] K. Huppler, "The art of building a good benchmark," in *Performance Evaluation and Benchmarking: Revised Selected Papers*, vol. 5895, Lyon, France: TPCTC, 2009, pp. 18–30. DOI: 10.1007/978-3-642-10424-4\textunderscore3. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-10424-4_3.
 - [18] M. Kjaer, M. Kihl, and A. Robertsson, "Resource allocation and disturbance rejection in web servers using SLAs and virtualized servers," *IEEE Transactions on Network and Service Management*, vol. 6, no. 4, pp. 226–239, Dec. 2009. DOI: 10.1109/tnsm.2009.04.090403. [Online]. Available: <https://doi.org/10.1109/tnsm.2009.04.090403>.
 - [19] R. Miller, *1&1 goes 'unlimited' with hosting bandwidth | data center knowledge | news and analysis for the data center industry*, Sep. 2009. [Online]. Available: <https://www.datacenterknowledge.com/archives/2009/09/03/11-goes-unlimited-with-hosting-bandwidth> (visited on 10/14/2022).
 - [20] J. Almeida, V. Almeida, D. Ardagna, Í. Cunha, C. Francalanci, and M. Trubian, "Joint admission control and resource allocation in virtualized servers," *Journal of Parallel and Distributed Computing*, vol. 70, no. 4, pp. 344–362, 2010, ISSN: 0743-7315. DOI: <https://doi.org/10.1016/j.jpdc.2009.08.009>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0743731509001567>.

- [21] D. Molnar and S. E. Schechter, "Self hosting vs. cloud hosting: Accounting for the security impact of hosting in the cloud.," in *WEIS*, vol. 2010, 2010, pp. 1–18.
- [22] A. Torrisi, *Free php benchmark performance script*, 2010. [Online]. Available: <http://www.php-benchmark-script.com/> (visited on 01/22/2023).
- [23] A. Lenk, M. Menzel, J. Lipsky, S. Tai, and P. Offermann, "What are you paying for? performance benchmarking for infrastructure-as-a-service offerings," in *2011 IEEE 4th International Conference on Cloud Computing*, 2011, pp. 484–491. DOI: 10.1109/CLOUD.2011.80.
- [24] A. Padilla and T. Hawkins, *Pro PHP Application Performance*. Apress, 2011. DOI: 10.1007/978-1-4302-2899-8. [Online]. Available: <https://doi.org/10.1007/978-1-4302-2899-8>.
- [25] T. Barker, *Pro JavaScript Performance*, en, 1st ed. Berlin, Germany: APress, Nov. 2012.
- [26] C. J. Hooper, N. Marie, and E. Kalampokis, "Dissecting the butterfly," in *Proceedings of the 4th Annual ACM Web Science Conference*, ACM, Jun. 2012. DOI: 10.1145/2380718.2380737. [Online]. Available: <https://doi.org/10.1145/2380718.2380737>.
- [27] Oracle Inc., *Brief history of virtualization*, 2012. [Online]. Available: https://docs.oracle.com/cd/E26996_01/E18549/html/VMUSG1010.html (visited on 09/30/2022).
- [28] F. Bazargan, C. Y. Yeun, and M. J. Zemerly, "State-of-the-art of virtualization, its security threats and deployment models," *International Journal for Information Security Research*, vol. 3, no. 3, pp. 335–343, 2013. DOI: 10.20533/ijisr.2042.4639.2013.0039.
- [29] I. Grigorik, *High Performance Browser Networking: What every web developer should know about networking and web performance*. O'Reilly Media, 2013, ISBN: 1449344763.
- [30] P. Pollock, *Web Hosting For Dummies*, 1st. For Dummies, 2013, ISBN: 1118540573.
- [31] P. Pollock, *Web hosting for dummies* (For dummies), en. Hoboken, New Jersey: John Wiley & Sons, Inc., May 2013.
- [32] R. Ahmed and R. Boutaba, *Collaborative Web Hosting*. Springer International Publishing, 2014. DOI: 10.1007/978-3-319-03807-0. [Online]. Available: <https://doi.org/10.1007/978-3-319-03807-0>.
- [33] D. Coghlan and M. Brydon-Miller, *The SAGE Encyclopedia of Action Research*. SAGE Publications Ltd, 2014. DOI: 10.4135/9781446294406. [Online]. Available: <https://doi.org/10.4135/9781446294406>.
- [34] T. Walkowiak, "Behavior of web servers in stress tests," in *Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. June 30 – July 4, 2014, Brunów, Poland*, Springer International

- Publishing, 2014, pp. 467–476. DOI: 10.1007/978-3-319-07013-1_45. [Online]. Available: https://doi.org/10.1007/978-3-319-07013-1_45.
- [35] A. Bartuskova and O. Krejcar, “Loading speed of modern websites and reliability of online speed test services,” in *Computational Collective Intelligence*, M. Núñez, N. T. Nguyen, D. Camacho, and B. Trawiński, Eds., Cham: Springer International Publishing, 2015, pp. 65–74, ISBN: 978-3-319-24306-1.
- [36] A. Ling, *Amazon.com: Practical Apache, PHP-FPM & Nginx Reverse Proxy: How to Build a Secure, Fast and Powerful Webserver from scratch (Practical Guide Series Book 3)*. Kindle Direct Publishing, 2015. [Online]. Available: <https://www.amazon.com/Practical-Apache-PHP-FPM-Nginx-Reverse-ebook/dp/B00X40K8M8>.
- [37] A. Bartuskova, O. Krejcar, T. Sabbah, and A. Selamat, “WEBSITE SPEED TESTING ANALYSIS USING SPEEDTESTING MODEL,” *Jurnal Teknologi*, vol. 78, no. 12-3, Dec. 2016. DOI: 10.11113/jt.v78.10028. [Online]. Available: <https://doi.org/10.11113/jt.v78.10028>.
- [38] M. Baruwat Chhetri, S. Chichin, Q. B. Vo, and R. Kowalczyk, “Smart cloudbench—a framework for evaluating cloud infrastructure performance,” *Information Systems Frontiers*, vol. 18, no. 3, pp. 413–428, 2016, ISSN: 1387-3326. DOI: 10.1007/s10796-015-9557-2.
- [39] W. Bervoets, *Fast, Scalable And Secure Web Hosting For Entrepreneurs, Learn to set up your server and website*, en. Hoboken, New Jersey: Leanpub, 2016.
- [40] A. Hussain, *Learning PHP 7 High Performance*. Birmingham, England: Packt Publishing, Jan. 2016.
- [41] T. Palit, Y. Shen, and M. Ferdman, “Demystifying cloud benchmarking,” in *2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, IEEE, Apr. 2016. DOI: 10.1109/ispass.2016.7482080. [Online]. Available: <https://doi.org/10.1109/ispass.2016.7482080>.
- [42] UpCloud, *How to benchmark cloud servers*, 2016. [Online]. Available: <https://upcloud.com/resources/tutorials/how-to-benchmark-cloud-servers> (visited on 10/06/2022).
- [43] R. Aley, *Pro functional PHP programming*, en, 1st ed. New York, NY: APRESS, Sep. 2017.
- [44] M. Allen, *The SAGE Encyclopedia of Communication Research Methods*. SAGE Publications, Inc, 2017. DOI: 10.4135/9781483381411. [Online]. Available: <https://doi.org/10.4135/9781483381411>.
- [45] I. Amazon Web Services, *Aws-refarch-wordpress/aws-refarch-wordpress-v20171026.jpeg at master · aws-samples/aws-refarch-wordpress · github*, 2017. [Online]. Available: <https://github.com/aws-samples/aws-refarch-wordpress/blob/master/images/aws-refarch-wordpress-v20171026.jpeg> (visited on 01/20/2023).
- [46] D. Beyer, S. Löwe, and P. Wendler, “Reliable benchmarking: Requirements and solutions,” *International Journal on Software Tools for Technology Transfer*,

- vol. 21, no. 1, pp. 1–29, Nov. 2017. DOI: 10.1007/s10009-017-0469-y. [Online]. Available: <https://doi.org/10.1007/s10009-017-0469-y>.
- [47] T. Butler, *NGINX Cookbook*. Birmingham, England: Packt Publishing, Aug. 2017.
- [48] P. Lorenc and M. Woda, “IaaS vs. traditional hosting for web applications - cost effectiveness analysis for a local market,” in *Advances in Dependability Engineering of Complex Systems*, Springer International Publishing, May 2017, pp. 233–243. DOI: 10.1007/978-3-319-59415-6_23. [Online]. Available: https://doi.org/10.1007/978-3-319-59415-6_23.
- [49] A. Rawdat, *Nginx rate limiting*, 2017. [Online]. Available: <https://www.nginx.com/blog/rate-limiting-nginx/> (visited on 01/19/2023).
- [50] R. Abela, *Wordpress_shared_hosting_setup.png (800×600)*, 2018. [Online]. Available: https://www.wpwhitesecurity.com/wp-content/uploads/2018/07/wordpress_shared_hosting_setup.png (visited on 01/18/2023).
- [51] R. Abela, *Wordpress_vps_hosting_setup.png (800×571)*, 2018. [Online]. Available: https://www.wpwhitesecurity.com/wp-content/uploads/2018/07/wordpress_vps_hosting_setup.png (visited on 01/18/2023).
- [52] D. Farinelli, *Server performance metrics: 8 you should be considering*, 2018. [Online]. Available: <https://raygun.com/blog/server-performance-metrics/> (visited on 10/01/2022).
- [53] V. Fedoseenko, *What is xaas? iaas vs saas vs paas: What's the difference. examples*, 2018. [Online]. Available: <https://www.ispsystem.com/news/xaas> (visited on 01/18/2023).
- [54] K. Bergener, N. Clever, and A. Stein, *Wissenschaftliches Arbeiten im Wirtschaftsinformatik-Studium*. Springer Berlin Heidelberg, 2019. DOI: 10.1007/978-3-662-57949-7. [Online]. Available: <https://doi.org/10.1007/978-3-662-57949-7>.
- [55] D. Beyer, S. Löwe, and P. Wendler, “Reliable benchmarking: Requirements and solutions,” *International Journal on Software Tools for Technology Transfer*, vol. 21, no. 1, pp. 1–29, 2019, ISSN: 1433-2779. DOI: 10.1007/s10009-017-0469-y.
- [56] V. Cassé, *Web hosting: How to host 3 million websites? - ovhcloud blog*, 2019. [Online]. Available: <https://blog.ovhcloud.com/web-hosting-how-to-host-3-million-websites/> (visited on 01/18/2023).
- [57] J. Lulka, “New hardware, use cases shape server benchmarking needs,” *TechTarget*, 2019. [Online]. Available: <https://www.techtarget.com/searchdatacenter/feature/New-hardware-use-cases-shape-server-benchmarking-needs> (visited on 09/29/2022).
- [58] S. Matic, G. Tyson, and G. Stringhini, “PYTHIA,” in *The World Wide Web Conference*, ACM, May 2019. DOI: 10.1145/3308558.3313664. [Online]. Available: <https://doi.org/10.1145/3308558.3313664>.
- [59] G. Schulz, *Data Infrastructure Management*. London, England: CRC Press, Jan. 2019.

- [60] P. Burgy, *A brief history of the content management system*, 2020. [Online]. Available: <https://opensource.com/article/20/7/history-content-management-system> (visited on 10/15/2022).
- [61] L. Busetto, W. Wick, and C. Gumbinger, "How to use and assess qualitative research methods," *Neurological Research and Practice*, vol. 2, no. 1, May 2020. DOI: 10.1186/s42466-020-00059-z. [Online]. Available: <https://doi.org/10.1186/s42466-020-00059-z>.
- [62] S. Döringer, "'the problem-centred expert interview'. combining qualitative interviewing approaches for investigating implicit expert knowledge," *International Journal of Social Research Methodology*, vol. 24, no. 3, pp. 265–278, May 2020. DOI: 10.1080/13645579.2020.1766777. [Online]. Available: <https://doi.org/10.1080/13645579.2020.1766777>.
- [63] I. Grigorik, P. Le Hégarret, and T. Reifsteck, **proposed* web performance working group charter*, 2020. [Online]. Available: <https://w3c.github.io/charter-webperf/> (visited on 01/21/2023).
- [64] S. M. Jain, *Linux Containers and Virtualization*, 1st ed. Berlin, Germany: APress, Oct. 2020.
- [65] J. Matson, *Php worker performance benchmarking and test results – pagely*, May 2020. [Online]. Available: <https://pagely.com/blog/php-worker-performance-benchmarking/> (visited on 01/23/2023).
- [66] J. Strebel, *Php workers and wordpress: The definitive guide – pagely*, May 2020. [Online]. Available: <https://pagely.com/blog/php-workers-wordpress-guide/> (visited on 01/23/2023).
- [67] S. Bisson, "This open-source microsoft benchmark is a powerful server testing tool," *TechRepublic*, 2021. [Online]. Available: <https://www.techrepublic.com/article/this-open-source-microsoft-benchmark-is-a-powerful-server-testing-tool/> (visited on 09/29/2022).
- [68] K. Ohashi, *Methodology - wordpress hosting performance benchmarks*, 2021. [Online]. Available: <https://wphostingbenchmarks.com/methodology/> (visited on 01/31/2023).
- [69] K. Ohashi and T. Kyle, *Kevin ohashi reveals how he discovers which wordpress hosting companies suck which do not - wpx blog: World's fastest wordpress host*, 2021. [Online]. Available: <https://blog.wpx.net/kevin-ohashi-reveals-how-he-discovers-which-wordpress-hosting-companies-suck-which-do-not/> (visited on 01/25/2023).
- [70] O. C. tkadlec, *Page-level metrics | webpagetest documentation*, 2021. [Online]. Available: <https://docs.webpagetest.org/metrics/page-metrics/> (visited on 01/27/2023).
- [71] T. C. tkadlec and stoyan, *Speed index | webpagetest documentation*, 2021. [Online]. Available: <https://docs.webpagetest.org/metrics/speedindex/> (visited on 01/27/2023).

- [72] S. Wang, K. MacMillan, B. Schaffner, N. Feamster, and M. Chetty, *A first look at the consolidation of dns and web hosting providers*, Oct. 8, 2021. [Online]. Available: <https://arxiv.org/pdf/2110.15345>.
- [73] A. Aleksandrov, *Wordpress hosting benchmark tool – wordpress plugin* | *wordpress.org*, 2022. [Online]. Available: <https://wordpress.org/plugins/wpbenchmark/> (visited on 01/23/2023).
- [74] A. Aleksandrov, *Wp-benchmark-io.php in wpbenchmark/trunk – wordpress plugin repository*, 2022. [Online]. Available: <https://plugins.trac.wordpress.org/browser/wpbenchmark/trunk/wp-benchmark-io.php#L487> (visited on 01/24/2023).
- [75] D. Carlo, *What's new in php 8 (features, improvements & the jit compiler)*, 2022. [Online]. Available: <https://kinsta.com/blog/php-8/> (visited on 01/25/2023).
- [76] 5. contributors, *Gatling - installation*, 2022. [Online]. Available: <https://gatling.io/docs/gatling/tutorials/installation/> (visited on 01/20/2023).
- [77] D. DeJonghe, *NGINX cookbook*, 2nd ed. Sebastopol, CA: O'Reilly Media, May 2022.
- [78] T. V. Doan, R. van Rijswijk-Deij, O. Hohlfeld, and V. Bajpai, "An empirical view on consolidation of the web," *ACM Transactions on Internet Technology*, vol. 22, no. 3, pp. 1–30, Aug. 2022. DOI: 10.1145/3503158. [Online]. Available: <https://doi.org/10.1145/3503158>.
- [79] G. D. Documentation, *About pagespeed insights* | *google developers*, 2022. [Online]. Available: <https://developers.google.com/speed/docs/insights/v5/about> (visited on 01/24/2023).
- [80] A. S. Foundation, *Apache jmeter - download apache jmeter*, 2022. [Online]. Available: https://jmeter.apache.org/download_jmeter.cgi (visited on 01/20/2023).
- [81] L. GoDaddy Operating Company, *Web hosting* | *lightning fast hosting & one click setup godaddy*, 2022. [Online]. Available: <https://www.godaddy.com/hosting/web-hosting> (visited on 10/15/2022).
- [82] T. P. Group, *Php: Php 8.0.0 release announcement*, 2022. [Online]. Available: <https://www.php.net/releases/8.0/en.php> (visited on 01/27/2023).
- [83] L. HostGator.com, *Shared website hosting - easy and affordable* | *hostgator*, 2022. [Online]. Available: <https://www.hostgator.com/web-hosting> (visited on 10/15/2022).
- [84] bluehost inc., *Wordpress hosting - fast, secure wp web hosting - bluehost*, 2022. [Online]. Available: <https://www.bluehost.com/wordpress/wordpress-hosting#pricing-cards> (visited on 10/15/2022).
- [85] I. Inc., *Web hosting* | *fast hosting services for \$0.50/month*, 2022. [Online]. Available: <https://www.ionos.com/hosting/web-hosting#plans> (visited on 10/15/2022).

- [86] K. Ohashi, *Benchmark.php in wpperformancetester/trunk – wordpress plugin repository*, 2022. [Online]. Available: <https://plugins.trac.wordpress.org/browser/wpperformancetester/trunk/benchmark.php> (visited on 01/24/2023).
- [87] K. Ohashi, *Wpperformancetester – wordpress plugin | wordpress.org*, 2022. [Online]. Available: <https://wordpress.org/plugins/wpperformancetester/> (visited on 01/23/2023).
- [88] C. U. Press, *Benchmarking | meaning, definition in cambridge english dictionary*, 2022. [Online]. Available: <https://dictionary.cambridge.org/dictionary/english/benchmarking> (visited on 10/16/2022).
- [89] I. Red Hat, *What is virtualization?* 2022. [Online]. Available: <https://www.redhat.com/en/topics/virtualization/what-is-virtualization> (visited on 10/05/2022).
- [90] D. Rose, *Managed WordPress Hosting 2022 » High-Performance aus Deutschland — hostpress.de*, 2022. [Online]. Available: <https://www.hostpress.de/blog/was-ist-managed-wordpress-hosting/> (visited on 01/08/2023).
- [91] H. Sandra, “Performance analysis of openlitespeed and apache web servers in serving client requests,” *Knowbase : International Journal of Knowledge in Database*, vol. 2, no. 2, pp. 114–129, 2022, ISSN: 2797-7501. [Online]. Available: <https://ejournal.uinbukittinggi.ac.id/index.php/ijokid/article/view/5306>.
- [92] B. Varghese, N. Wang, D. Bermbach, C.-H. Hong, E. de Lara, W. Shi, and C. Stewart, “A survey on edge performance benchmarking,” *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–33, 2022, ISSN: 0360-0300. DOI: 10.1145/3444692.
- [93] C. Xilogianni, F.-R. Doukas, I. C. Drivas, and D. Kouis, “Speed matters: What to prioritize in optimization for faster websites,” *Analytics*, vol. 1, no. 2, pp. 175–192, Nov. 2022. DOI: 10.3390/analytics1020012. [Online]. Available: <https://doi.org/10.3390/analytics1020012>.
- [94] L. Zembruzki, R. Sommese, L. Z. Granville, A. S. Jacobs, M. Jonker, and G. C. M. Moura, “Hosting industry centralization and consolidation,” in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, IEEE, Apr. 2022. DOI: 10.1109/noms54207.2022.9789881. [Online]. Available: <https://doi.org/10.1109/noms54207.2022.9789881>.
- [95] A. Aleksandrov, *Wordpress hosting speed benchmark*, 2023. [Online]. Available: <https://wpbenchmark.io/> (visited on 01/23/2023).
- [96] Atlantic.Net, *What is colocation hosting in 2022? benefits of colocation hosting*. 2023. [Online]. Available: <https://www.atlantic.net/orlando-colocation-hosting-data-center/what-is-colocation-hosting/> (visited on 01/18/2023).
- [97] I. Gartner, *Definition of infrastructure as a service (iaas) - gartner information technology glossary*, 2023. [Online]. Available: <https://www.gartner.com/en/information-technology/glossary/infrastructure-as-a-service-iaas> (visited on 01/17/2023).

- [98] I. Gartner, *Definition of web hosting - gartner information technology glossary*, 2023. [Online]. Available: <https://www.gartner.com/en/information-technology/glossary/web-hosting> (visited on 01/18/2023).
- [99] P. I. GmbH, *Hosting control panel - hosting wikipedia*, 2023. [Online]. Available: <https://www.plesk.com/wiki/hosting-control-panel/> (visited on 01/21/2023).
- [100] GTmetrix, *Gtmetrix | website performance testing and monitoring*, 2023. [Online]. Available: <https://gtmetrix.com/> (visited on 01/24/2023).
- [101] GTmetrix, *Gtmetrix test server locations | gtmetrix*, 2023. [Online]. Available: <https://gtmetrix.com/locations.html> (visited on 01/24/2023).
- [102] C. Inc., *Cloudlinux os components | documentation*, 2023. [Online]. Available: https://docs.cloudlinux.com/cloudlinux_os_components/ (visited on 01/22/2023).
- [103] S. Inc., *Debian – informationen über quellcode-paket php-defaults in sid*, 2023. [Online]. Available: <https://packages.debian.org/de/source/sid/php-defaults> (visited on 01/22/2023).
- [104] T. International, *About us*, 2023. [Online]. Available: https://www.webhosting-performance.com/en/about_us (visited on 01/22/2023).
- [105] T. International, *Benchmark results | webhosting performance*, 2023. [Online]. Available: https://www.webhosting-performance.com/en/benchmark_results (visited on 01/22/2023).
- [106] T. International, *Benchmark test your website | webhosting performance*, 2023. [Online]. Available: <https://www.webhosting-performance.com/en/download> (visited on 01/22/2023).
- [107] G. Labs, *What is load testing? how to create a load test in k6*, 2023. [Online]. Available: <https://k6.io/docs/test-types/load-testing/> (visited on 01/25/2023).
- [108] G. Labs, *What is stress testing? how to create a stress test in k6*, 2023. [Online]. Available: <https://k6.io/docs/test-types/stress-testing/> (visited on 01/25/2023).
- [109] R. Ltd., *Redis*, 2023. [Online]. Available: <https://redis.io/> (visited on 01/20/2023).
- [110] K. Ohashi, *<\$25/month wordpress hosting performance benchmarks 2022*, 2023. [Online]. Available: <https://wphostingbenchmarks.com/benchmark/2022-25-wordpress/> (visited on 02/07/2023).
- [111] K. Ohashi, *Enterprise (\$500+/month) wordpress hosting performance benchmarks 2022*, 2023. [Online]. Available: <https://wphostingbenchmarks.com/benchmark/2022-enterprise-wordpress/> (visited on 02/07/2023).
- [112] Q-Success, *Historical trends in the usage statistics of content management systems, january 2023*, 2023. [Online]. Available: https://w3techs.com/technologies/history_overview/content_management/all (visited on 01/09/2023).

- [113] Q-Success, *Usage statistics and market share of content management systems, january 2023*, 2023. [Online]. Available: https://w3techs.com/technologies/overview/content_management (visited on 01/09/2023).
- [114] Q-Success, *Usage statistics and market share of javascript libraries for websites, january 2023*, 2023. [Online]. Available: https://w3techs.com/technologies/overview/javascript_library (visited on 01/08/2023).
- [115] Q-Success, *Usage statistics and market share of server-side programming languages for websites, january 2023*, 2023. [Online]. Available: https://w3techs.com/technologies/overview/programming_language (visited on 01/08/2023).
- [116] Q-Success, *Usage statistics of css for websites, january 2023*, 2023. [Online]. Available: <https://w3techs.com/technologies/details/ce-css> (visited on 01/08/2023).
- [117] 15th ACM Web Science Conference 2023 April 30th May 1st. Online and in person. Co-located with The Web Conference, Austin, Texas, USA — websci23.webscience.org. [Online]. Available: <https://websci23.webscience.org/> (visited on 12/27/2022).
- [118] I. Amazon Web Services, *What is Web Hosting? - Web Hosting Explained - AWS* — [aws.amazon.com](https://aws.amazon.com/what-is/web-hosting/). [Online]. Available: <https://aws.amazon.com/what-is/web-hosting/> (visited on 12/28/2022).
- [119] 7. contributors, *Installation — locust 2.14.2 documentation*. [Online]. Available: <https://docs.locust.io/en/stable/installation.html#installation> (visited on 01/20/2023).
- [120] Dormando, *Memcached - a distributed memory object caching system*. [Online]. Available: <https://memcached.org/>.
- [121] I. Red Hat, *What's the difference between cloud and virtualization?* — [redhat.com](https://www.redhat.com/en/topics/cloud-computing/cloud-vs-virtualization). [Online]. Available: <https://www.redhat.com/en/topics/cloud-computing/cloud-vs-virtualization> (visited on 12/28/2022).

YOUR KNOWLEDGE HAS VALUE



- We will publish your bachelor's and master's thesis, essays and papers
- Your own eBook and book - sold worldwide in all relevant shops
- Earn money with each sale

Upload your text at www.GRIN.com
and publish for free

